Appariement de descripteurs évoluant dans le temps : application à la comparaison d'assurance

Anne-Lise Bedenel

Université de Lille
Inria Lille Nord Europe
MeilleureAssurance.com

École doctorale **ED Régionale SPI 72**
Unité de recherche **Equipe-projets MΘDAL**

Thèse présentée par **Anne-Lise Bedenel**

Soutenue le **3 avril 2019**

En vue de l'obtention du grade de docteur de l'Université de Lille et de l'Inria Lille Nord Europe

Discipline **Mathématiques**
Spécialité **Statistique**

Titre de la thèse

Appariement de descripteurs évoluant dans le temps

Application à la comparaison d'assurance

Thèse dirigée par	Christophe Biernacki	co-directeur
	Laetitia Jourdan	co-directrice

Composition du jury

Rapporteurs	Faicel Chamroukhi	professeur à l'Université de Caen	
	Frédéric Saubion	professeur à l'Université d'Angers	
Examinateurs	Cristian Preda	professeur à l'université de Lille	président du jury
	Jean-Charles Boisson	mcf à l'univesité de reims	
Invités	Christophe Triquet		
	Emmanuel De Castro		
Directeurs de thèse	Christophe Biernacki	professeur à l'université de Lille	
	Laetitia Jourdan	professeure à l'université de Lille	

Cette thèse a été préparée dans les laboratoires suivants.

Equipe-projets MΘDAL
40 avenue Halley
59650 Villeneuve d'ascq Cedex
France

☎ (33)(0)3 59 57 78 00
Site https://modal.lille.inria.fr/

Equipe-projets ORKAD
Université Lille, M3
Avenue Carl Gauss
59650 Villeneuve d'ascq Cedex
France

☎ (33)(0)3 28 77 85 41
Site https://www.cristal.univ-lille.fr/

Laboratoire Paul Painlevé
CNRS U.M.R 8524
59655 Villeneuve d'ascq Cedex
France

☎ (33)(0)3 20 43 48 50
Site https://math.univ-lille1.fr/

Appariement de descripteurs évoluant dans le temps
Application à la comparaison d'assurance

Résumé

Dans le domaine de la comparaison d'assurances en ligne, les données évoluent constamment, impliquant certaines difficultés pour les exploiter. En effet, la plupart des méthodes d'apprentissage standards, comme la classification supervisée, nécessitent des descripteurs de données identiques pour les échantillons d'apprentissage et de test. Or, les formulaires en lignes d'où proviennent les données sont régulièrement modifiés, impliquant de travailler avec une faible quantité de données. L'objectif est alors d'utiliser les données obtenues avant la modification des descripteurs pour générer de nouveaux échantillons et augmenter la taille des échantillons observés après la modification. Nous proposons donc d'effectuer un transfert de connaissances entre les données observées avant et après la modification des variables. Les données étant observées soit avant, soit après la modification de la variable, entraînent un problème de données manquantes où les liens entre les descripteurs avant et après la modification sont totalement inconnus. Une modélisation probabiliste du problème, modélisant la loi jointe de la variable avant et après la modification de ses descripteurs est proposée. Le problème revient alors à un problème d'estimation dans un graphe où le modèle n'est pas identifiable. L'identifiabilité du modèle est assurée par des contraintes métiers et techniques, amenant à travailler avec un ensemble réduit de modèles très parcimonieux. Deux méthodes d'estimation des paramètres reposant sur des algorithmes EM sont proposées : une estimation par vraisemblance profilée et une estimation jointe des paramètres. L'ensemble de modèles amène à une étape de sélection de modèle, effectuée selon deux critères : un critère asymptotique et un critère non asymptotique reposant sur l'analyse bayésienne, comprenant une stratégie d'échantillonnage préférentiel combinée à un algorithme de Gibbs. Pour obtenir la méthode ayant le meilleur compromis "résultats-temps de calcul", une recherche exhaustive (EXsearch) et une recherche non-exhaustive (AGsearch) sont comparées. La recherche AGsearch, basée sur un algorithme génétique, combine à la fois l'estimation (problème continu) et la sélection de modèle (problème combinatoire). La thèse se termine par une comparaison des méthodes et critères proposés afin d'obtenir une stratégie optimale, puis par une application sur des données réelles.

Mots clés : transfert de connaissance, sélection de modèle, algorithme génétique

Abstract

In the online insurance comparison field, data constantly evolve, implying some difficulties to exploit them. Indeed, most of the classical learning methods, as supervised classification, require data descriptors equal to both learning and test samples. Online forms where data come from are often changed. These constant modifications of data descriptors lead us to work with the small amount of data and make analysis more complex. So, the goal is to use data generated before the feature descriptors modification. By doing so, we generate new samples and increase the size of the observed sample after the descriptors modification. We intend to perform a learning transfer between observed data before and after features modification. Data are observed either before, or after feature modification which bring a problem of missing data. Also, the links between data descriptors of the feature before and after the modification are totally unknown. A probabilistic modelling of the problem has been suggested to modelize the joint distribution of the feature before and after the modification of the data descriptors. The problem becomes an estimation problem in a graph where the model is unidentifiable. Some business and technical constraints ensure the identifiability of the model and we have to work with a reduced set of very parsimonious models. Two methods of estimation rely on EM algorithms have been intented. The first one is an estimation by profile likelihood and the second one is a join estimation of parameters. The constraints set lead us to work with a set of models. A model selection step is required. For this step, two criterium are proposed: an asymptotic criterium and a non-asymptotic criterium rely on Bayesian analysis which includes an importance sampling combined with Gibbs algorithm. To have an optimal method for both results and execution time, two research strategies are suggested. The first strategy (EXsearch) is an exhaustive search and the second strategy (AGsearch) is a non-exhaustive search based on genetic algorithm, combining both estimation (continuous problem) and selection (combinatorial problem). This thesis finishes with a comparison of methods and criteria proposed to detect the optimal strategy in a business framework and with an application on real data.

Keywords: transfer learning, models selection, genetic algortithms

Remerciements

Mon parcours en doctorat a été optimisé par une combinaison de nombreuses personnes ayant permis la réussite de cette thèse. Il me parait alors logique de commencer ce mémoire en les remerciant .

Je tiens tout d'abord à remercier Christophe Biernacki et Laetitia Jourdan pour m'avoir encadrée durant toute la durée de cette thèse. Je tiens à leur adresser mes remerciements notamment pour leur accompagnement, leurs conseils précieux et surtout leur patience.

Je souhaite, également, remercier Faicel Chamroukhi et Frédéric Saubion d'avoir accepté le rôle de rapporteur pour mon travail. Je remercie également les autres membres du jury Cristian Preda et Jean-Charles Boisson pour avoir accepté le rôle d'examinateur.

Cette thèse n'aurait jamais eu lieu, sans la société MeilleurAssurance et notamment son Directeur Christophe Triquet. Je le remercie infiniment pour m'avoir donné l'opportunité ainsi que les moyens d'effectuer cette thèse dans les meilleurs conditions. Pour cette raison, je remercie également Emmanuel De Castro pour m'avoir suivie tout au long de cette thèse.

Je remercie également, les deux équipes (ORKAD et MODAL) dans lesquelles j'ai travaillé, pour leur accueil, leur soutient et leurs conseils et notamment les doctorants d'ORKAD et MODAL : Maxime, Aymeric, Lucien, Quentin et Adrien pour les moments passés ensembles.

Je tiens également à remercier l'ensemble de mes collègues (anciens et actuels) de MeilleurAssurance pour leur soutient tout au long de cette thèse et leur intérêt porté à mon travail malgré le coté abstrait de la recherche, notamment Nina, pour ses conseils de coureuse longue distance pouvant s'appliquer, au final, à un parcours de doctorat.

Cette thèse n'aurait sans doute jamais aboutie sans les encouragements, le soutien, les nombreuses soirées et moments de détente partagés avec mes amis proches : Rose, Juliette, Jérôme, Julie, GuiGui, Lulie, Adé, Shout, Cook, Aymeric, Maxime, Olivier, Johann, Léa, Morgane, Amel, Céline, Marion, la Colloc et tous ceux qui ont croisé mon chemin

lors de ces moments. J'adresse tout particulièrement un grand merci à Maxime pour son aide précieuse en JAVA me permettant ainsi un gain de temps significatif, à Guigui pour son temps passé à corriger les fautes d'orthographe de ce mémoire, à Johann pour son aide en anglais et Julie pour nos déjeuners réguliers qui m'ont fait beaucoup de bien.

Je remercie également ma famille au sens large : Grand-parents, oncles, tantes, ,cousins, petits-cousins, filleules, Mimi, Catherine, ... pour leur soutien, leur affection, leurs encouragements et tous les moments en famille m'apportant à chaque fois énormément de joie et de réconfort. Je continue en remerciant sincèrement ma sœur pour sa présence tout au long de cette thèse, sa confiance, son soutient et pour m'avoir offert son plus beau cadeau, ma filleule Louise qui est un concentré de bonheur à elle toute seule. Je tiens également à remercier mes parents qui ont toujours cru en moi, et qui ont toujours tout fait pour me donner les moyens de réussir.

Ces remerciements s'achèvent par des remerciements tout particuliers à celui qui partage ma vie depuis maintenant 5 ans. Je le remercie pour m'avoir encouragé à commencer cette thèse, pousser à ne jamais abandonner malgré les moments de stress, de doute où je n'ai pas été forcément facile à vivre. Je le remercie également pour ses heures passées à corriger et à m'écouter répéter mes présentations. Enfin, je le remercie d'avoir été attentionné, affectueux et là tout simplement.

Mon dernier remerciement s'adresse à mon chat, toujours présent pour dormir sur mon clavier lors mes longues soirées de rédaction.

Acronymes

A | B | E | I | J | L | M | R | S

A

AG Algorithme Génétique. 118
AGsearch Stratégie de recherche par Algorithme Génétique. 118
AIC Akaike Information Criterion. 75
AN Affaire Nouvelle. 10

B

BIC Bayesian Information Criterion. 75
BIL Bayesian Integrated Likehood. 75
BMA Bayesian Model Averaging. 96

E

EM Expectation-Maximisation. 60
EMV Estimateur du Maximum de Vraisemblance. 82
EXsearch Stratégie de recherche Exhaustive. 73

I

ITI Incremental Tree Induction. 31

J

JDS Journée de Statistiques. 84

L

LION Learning and Intellingent Optimization Conference. 105

M

MCMC Markov chain Monte Carlo. 93
MER Mise en relation. 7

MV Maximum de Vraisemblance. 58

R

ROADEF Société Française de Recherche Opérationnelle et d'Aide à la Décision. 105

S

SBX Simulated Binary Crossover. 122

Symboles

A Échantillon d'apprentissage 21
$\mathbf{A}_p$ Matrice de taille de $m \times q$ 169
$\mathcal{D}$ Un domaine 32
d Dimension de l'espace $\mathcal{X}$ 20
$\mathcal{D}_\mathcal{C}$ Un domaine cible 33
D_C Données du domaine cible 33
Δ Matrice de taille de $m^* m$ indiquant les contraintes posées sur le modèles 55, 170
$\boldsymbol{\delta}$ Un modèle 75
$\mathcal{D}_\mathcal{S}$ Un domaine source 33
D_S Données du domaine source 32
g Nombre de classe 20
I Fonction d'importance de l'échantillonnage préférentiel 92
i Indice de l'individu dans l'échantillon de taille n 21
I^* Fonction d'importance idéale pour l'échantillonnage préférentiel 93
$\hat{I}^*$ Fonction d'importance idéale estimée pour l'échantillonnage préférentiel 93
$\boldsymbol{\Delta}$ Un ensemble de modèles 72, 75
m Taille du vecteur $\mathbf{p}_.$ 169
n Nombre d'individus de l'échantillon 17, 48
n_h Nombre d'observations pour la modalité h 90
$n_{hh'}$ Nombre d'observations pour les modalités h et h' 90
n^- Nombre d'individus de l'échantillon avant modification 17, 48
n^+ Nombre d'individus de l'échantillon après modification 17, 48
ν Nombre de paramètres libres d'un modèle 82
$\mathcal{P}$ Espace des paramètres $\mathbf{p}$ 88
$\mathcal{P}_.$ Espace des paramètres $\mathbf{p}_.$ 53
p Nombre de modalités de la variable $\boldsymbol{x}$ 17

Table des matières

Liste des tableaux

Table des figures

Introduction générale

Le travail présenté dans cette thèse est issue d'une problématique proposée par la société *MeilleureAssurance.com*, finançant cette thèse CIFRE. *MeilleureAssurance.com* est une filiale de la société MeilleurTaux.com. *MeilleureAssurance.com* est un comparateur d'assurances en ligne dont l'objectif est de proposer à ses internautes les meilleures offres selon leurs attentes et leurs profils. Cette thèse s'inscrit dans la cadre d'une collaboration avec Inria (centre - Inria Lille - Nord Europe) et plus précisément les équipes-projets MΘDAL (MΘdel for Data Analysis and Learning) et ORKAD (Operationnal Reseach, Knowledge and Data). L'équipe MΘDAL est spécialisée dans le domaine de l'apprentissage statistique et l'équipe ORKAD travaille sur des méthodes exploitant simultanément des méthodes d'optimisation combinatoire et d'extraction de connaissances afin de résoudre des problèmes d'optimisation.

L'objectif d'un comparateur d'assurances est de proposer à ses internautes l'offre la plus adaptée à leurs attentes, selon leurs profils. Pour la plupart des comparateurs d'assurances en ligne, la comparaison se fait sur un seul critère : le prix. Afin d'affiner la comparaison des internautes, la société souhaite créer un modèle permettant de prédire l'offre la plus en adéquation avec les attentes de l'internaute, indépendamment du prix. Cet objectif en statistique est plutôt classique mais le fonctionnement d'un comparateur d'assurances en ligne a des spécificités empêchant d'appliquer directement les méthodes classiques. Pour réaliser une comparaison en ligne, un internaute doit remplir un formulaire de questions. Ce formulaire reprend les questions des systèmes de tarifications des assureurs partenaires du comparateur. Ainsi, à l'aide d'un web service, les assureurs partenaires peuvent renvoyer à l'internaute le prix réel de l'offre selon le profil qu'il a renseigné et ses attentes. La particularité d'un comparateur d'assurance vient du fait que les formulaires proposés aux internautes changent constamment. Le comparateur d'assurances adapte son formulaire en permanence pour deux raisons principales :

— *Pour les assureurs* : chaque assureur ayant son propre système de tarification, les

questions ne sont donc pas homogènes entre tous les assureurs partenaires.
— *Pour les internautes* : les questions sont adaptées régulièrement afin de viser plus de clarté et/ou de simplicité.

Ces changements peuvent être des ajouts, des suppressions, des fusions ou des modifications de questions. Avec ces modifications de variables, créer, par exemple, un modèle de classification supervisée devient alors complexe. La plupart des méthodes statistiques standards requièrent des échantillons conséquents pour l'apprentissage et s'appuient sur des descripteurs identiques entre les échantillons d'apprentissage et de test. Ces conditions ne sont alors pas remplies avec les données provenant du comparateur d'assurances.

À travers cette thèse, l'objectif de *MeilleureAssurance.com* est de réaliser un modèle prédictif prenant en compte les attentes et le profil des internautes mais également de comprendre les impacts de ses modifications sur le comportement des internautes. Par exemple, "Le changement de descripteur a-t-il eu l'impact voulu ?" ou "Comment les internautes ont-ils réagi face à ce changement ?". L'objectif de cette thèse consiste alors à apparier les descripteurs de variables qui ont été modifiés afin de pouvoir réaliser des analyses statistiques robustes face à ces changements mais également afin de pouvoir comprendre l'impact de ces modifications sur les internautes et leur comportement.

Ce mémoire se décompose en 6 chapitres :

Le **Chapitre 1** décrit tout d'abord le contexte dans lequel s'inscrit cette thèse. Il présente dans un premier temps, l'entreprise *MeilleureAssurance.com* et reprend la problématique de la société. Afin de bien comprendre les difficultés liées au domaine, ce chapitre présente dans un second temps le domaine de la comparaison d'assurances et les multiples spécificités qui lui sont liées. Enfin, dans un troisième temps, les verrous scientifiques associés à ces caractéristiques sont présentés.

Le **Chapitre 2** présente les différentes approches possibles pour réaliser un modèle de classification. L'objectif initial de cette thèse étant la réalisation d'un modèle de classification, dans un premier temps les méthodes de classification standards ainsi que les méthodes de classification évolutives de la littérature statistique traitant du problème sont présentées. Dans un second temps, nous nous intéresserons au transfert de connaissances qui sera également présenté. Enfin, le transfert de connaissances hétérogènes ainsi que différentes méthodes existantes dans la littérature de différentes communautés

(Statistique, Machine learning, ..) y seront plus particulièrement détaillées.

Le **Chapitre 3** présente la modélisation probabiliste proposée et ses caractéristiques. Ce chapitre se décompose en trois parties. La première partie concerne la modélisation probabiliste adaptée pour le problème, basée sur la loi jointe des données. Le problème comportant de nombreuses données manquantes, la modélisation proposée doit être adaptée pour avoir un modèle identifiable. La seconde partie de ce chapitre concerne l'identifiabilité en paramètre du modèle. Dans cette partie, des contraintes sont proposées amenant à travailler avec un ensemble de modèles. La troisième partie de ce chapitre concerne l'estimation des paramètres des modèles, à travers une présentation de la méthode utilisée et son application sur les modèles.

Les contraintes proposées dans la seconde partie du chapitre 3 impliquent de travailler avec un ensemble de modèles. L'objet du **Chapitre 4** concerne la sélection de modèle. Ce chapitre se décompose en 4 parties. La première partie concerne l'identifiabilité des modèles, deux modèles différents pouvant donner des résultats identiques. La seconde partie présente le critère asymptotique utilisé pour la sélection de modèle, le critère BIC. Les limites de ce critère étant rapidement atteintes avec la modélisation proposée, un second critère est proposé. Une troisième partie détaillera le critère **BIL**, non asymptotique que nous proposons. Dans une quatrième partie, les performances du critère **BIL** seront étudiées et une comparaison des critères BIC et **BIL** sera effectuée.

La méthode **EXsearch** présentée dans les chapitres 3 et 4 est une méthode exhaustive. Cette méthode nécessite l'estimation et la comparaison de l'ensemble des modèles. Le temps de calcul de la méthode peut vite devenir très coûteux. Dans le **Chapitre 5**, nous proposons d'utiliser une recherche non-exhaustive, basée sur un algorithme génétique afin d'élargir l'espace de recherche et réduire les temps de calculs. Ce chapitre présente, dans un premier temps, les méthodes d'optimisation existantes. Dans un second temps, la stratégie **AGBIC** est détaillée. Cette stratégie se base sur un algorithme génétique, adaptée à la modélisation probabiliste proposée dans le chapitre 3.

Dans le **Chapitre 6** les différentes stratégies combinant méthodes de recherche et critères de sélection de modèles sont comparées afin de trouver la stratégie permettant le meilleur compromis "rapidité-résultats". Une première comparaison est alors effectuée sur les performances des critères de sélection. Dans un second temps, les méthodes **EXsearch** et **AGsearch** sont comparées. Ces deux méthodes sont comparées à travers

plusieurs critères : les valeurs minimisant le critère de sélection pour les divers jeux de données, leurs temps de calcul et la précision des valeurs des paramètres estimés. Par ailleurs, la société MeilleureAssurance.com avait pour objectifs initiaux : la réalisation d'un modèle de classification robuste face aux changements des variables et l'obtention de modèles interprétables d'un point de vue métier. Une dernière partie se consacre donc à l'interprétabilité des modèles proposés par les stratégies et leurs performances dans un modèle de classification.

Ce mémoire se termine par une conclusion qui reprend nos contributions et ouvre sur différentes perspectives.

Chapitre 1

Contexte et objectif

> Donner une explication simplifiée à une problématique complexe ne signifie pas que sa résolution est simple.
>
> *Nanan-akassimandou, 1975*

Dans ce chapitre, la problématique de la société à l'origine de ce travail, *MeilleureAssurance.com*, un comparateur d'assurance en ligne, est exposée. Plutôt que de proposer aux internautes une liste d'offres triées par prix, l'objectif de la société est de proposer à ses internautes des offres adaptées à leurs attentes et à leurs profils. L'objectif revient alors à réaliser un modèle permettant la prédiction des offres les plus pertinentes pour les internautes. Le problème pour la réalisation de ce modèle est lié aux spécificités du domaine du web et de la comparaison d'assurances. Ces particularités sont également détaillées dans ce chapitre. Dans un dernier temps, nous présentons les verrous scientifiques impliqués dans ce travail.

1.1 Problématique de MeilleureAssurance.com

Dans cette section, nous introduisons le cadre dans lequel s'inscrit notre travail au travers de la problématique soulevée par la société *MeilleureAssurance.com*. Nous présentons rapidement la société, pour ensuite détailler les spécificités liées au domaine de la comparaison d'assurances et enfin la problématique de la société.

1.1.1 MeilleureAssurance.com

La société MeilleureAssurance.com est un comparateur d'assurance en ligne. Fondée en 2010 par Christophe TRIQUET, cette société lilloise est récemment devenue une

filiale du groupe MeilleurTaux. Le groupe MeilleurTaux propose actuellement de nombreux services financiers en ligne et la société MeilleureAssurance.com gère toute la partie comparaison d'assurances en ligne. Actuellement, MeilleureAssurance.com est le 3ème comparateur d'assurances en France derrière le Lynx et les Furets, ses principaux concurrents.

L'objectif principal d'un comparateur d'assurances est de permettre aux internautes venant sur son site, de comparer différentes offres afin de pouvoir trouver l'offre la plus adaptée à leurs besoins. La société donne la possibilité à ses internautes de comparer différents produits d'assurances telles que de la mutuelle santé, de l'assurance automobile, de l'assurance habitation ou encore de l'assurance animaux. De plus, la société propose différents sites pour réaliser une comparaison, notamment le site historique de la société : **Lecomparateurassurance.com** et un site plus récent : **Meilleureassurance.com**. Actuellement, ces sites génèrent plus de 3 millions de visiteurs par mois. Pour se distinguer de ses principaux concurrents, MeilleureAssurance.com propose également un service de proximité. Ce service permet aux internautes de comparer le prix des offres proposées par des assureurs ayant des agences près de chez eux, grâce à un système de géolocalisation. Ainsi, les internautes ont la possibilité de se rendre en agence si nécessaire. Bien que ses clients soient les assureurs et courtiers partenaires, la société MeilleureAssurance.com, met également au cœur de ses préoccupations les internautes et l'expérience utilisateurs. Un comparateur d'assurance est totalement gratuit pour les internautes, effectivement, ce sont les assureurs qui rémunèrent le comparateur selon différents modèles économiques qui seront détaillés dans la section 1.1.3. Le comparateur a un rôle d'intermédiaire entre les internautes et les assureurs partenaires. En effet, le comparateur met en relation les internautes venant faire une comparaison et les assureurs. Suite à cette mise en relation, les assureurs rémunèrent le comparateur. Les clients du comparateur sont donc les assureurs et non les internautes. Plus l'internaute est intéressé par l'offre d'assurance, plus la rémunération payée par l'assureur sera élevée. Trois niveaux de rémunération sont distingués. L'objectif du comparateur est alors double : proposer aux internautes des offres de qualités et pertinentes, afin que ceux-ci soient les plus intéressés possibles et avoir le plus d'internautes pertinents possibles à mettre en relation avec les assureurs.

1.1.2 Problématique initiale

De façon générale, un comparateur d'assurances propose aux internautes venant sur son site une comparaison des différentes offres selon un unique critère : le prix. La

société MeilleureAssurance.com souhaite se différencier en proposant la(les) meilleure(s) offre(s), non pas en fonction du prix uniquement, mais également en fonction des attentes et du profil de l'internaute. En effet, la société s'est rendu compte que certains internautes étaient plus intéressés par une offre de qualité, que par une offre peu coûteuse. L'idée est alors d'accroître la pertinence des offres proposées aux internautes. Cet objectif, qui revient à un problème de classification, est assez classique et de nombreuses méthodes de classifications supervisées tels que les arbres de décision ou la régression logistique pourraient y répondre. Cependant, le domaine du web, et plus particulièrement, celui de la comparaison d'assurances ont quelques spécificités qui empêchent l'utilisation de ces méthodes standards.

1.1.3 La comparaison d'assurances en ligne

Pour faire une comparaison d'assurances en ligne, un internaute remplit un formulaire en ligne tel que celui de la figure 1.1. Celui-ci contient un certain nombre de questions concernant le profil, le projet et les souhaits de l'internaute. Lorsque le formulaire est rempli et validé, les informations collectées sont envoyées à l'aide d'un web-service aux différents assureurs partenaires de la société. Ceux-ci renvoient alors, à leur tour, les tarifs des produits correspondant au profil et à la demande de l'internaute. Les différents tarifs des assureurs partenaires sont ensuite affichés à l'internaute, comme le montre la figure 1.2. Cette page est appelée "page de restitution". Sur cette page, l'internaute a ensuite deux possibilités.

Fiche Les offres et tarifs restitués ne lui conviennent pas, il quitte la page. Dans ce cas, ses coordonnées sont sauvegardées et un algorithme traite le profil de l'internaute pour le mettre en relation avec un ou plusieurs assureurs non proposés sur la page de restitution. Ce type de mise en relation est alors défini comme étant une **Fiche** et l'internaute sera défini comme *peu qualifié*. 65% des internautes sont vendus en tant que fiches.

MER Une ou plusieurs offres conviennent à l'internaute qui demande à être mis en relation avec les assureurs correspondant à son interêt. Ce type de relation est alors définie comme étant une **MER**. L'internaute étant à l'initiative de la demande de mise en relation sera défini comme *qualifié*. Environ 35% des internautes sont vendus en tant que MER.

Lors d'une MER, l'internaute choisit le ou les assureurs avec lequel il souhaite être mis en relation. Lors d'une fiche, un algorithme choisit les assureurs avec lesquels l'internaute sera mis en relation. Dans les deux cas, une mise en relation entre l'assureur

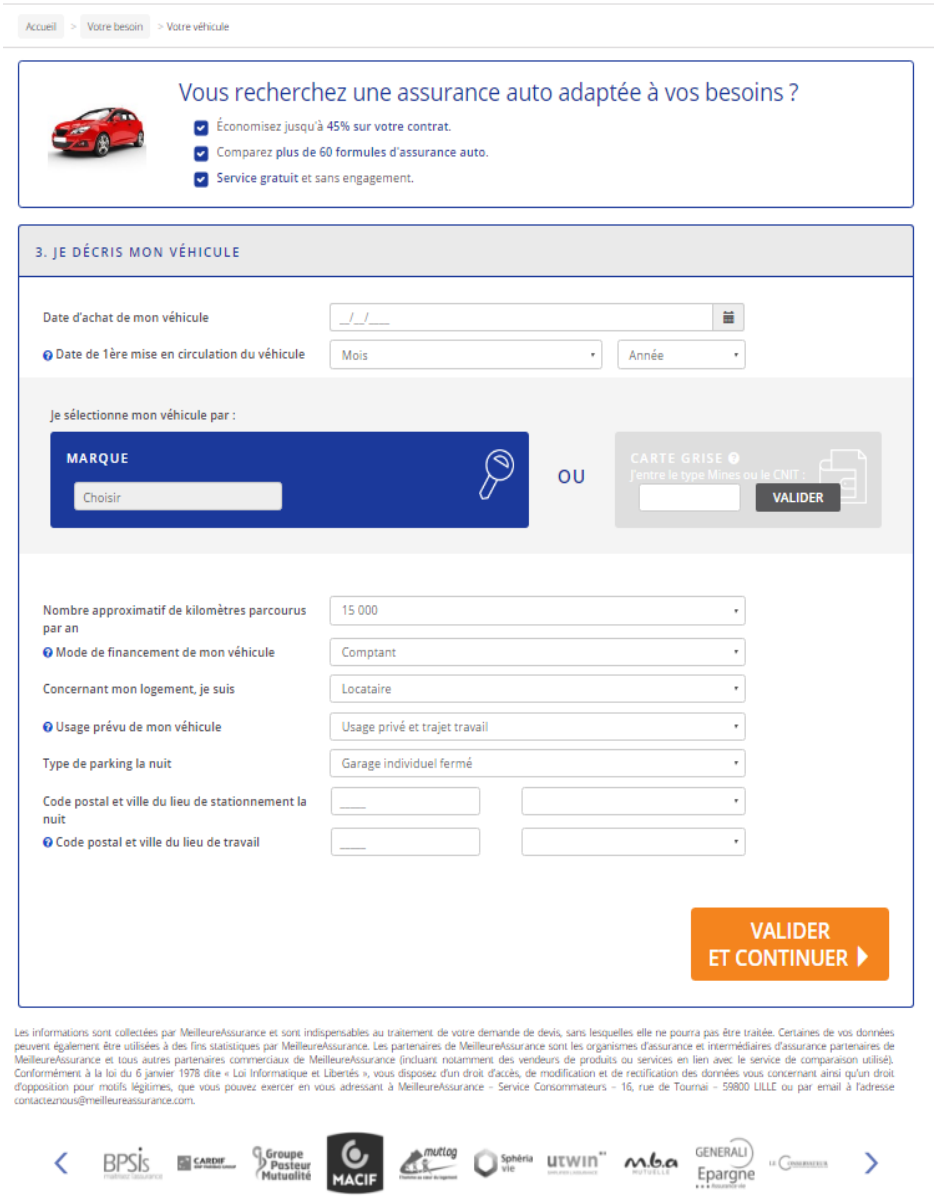


FIGURE 1.1 – Exemple de formulaire

FIGURE 1.2 – Exemple de page de restitution

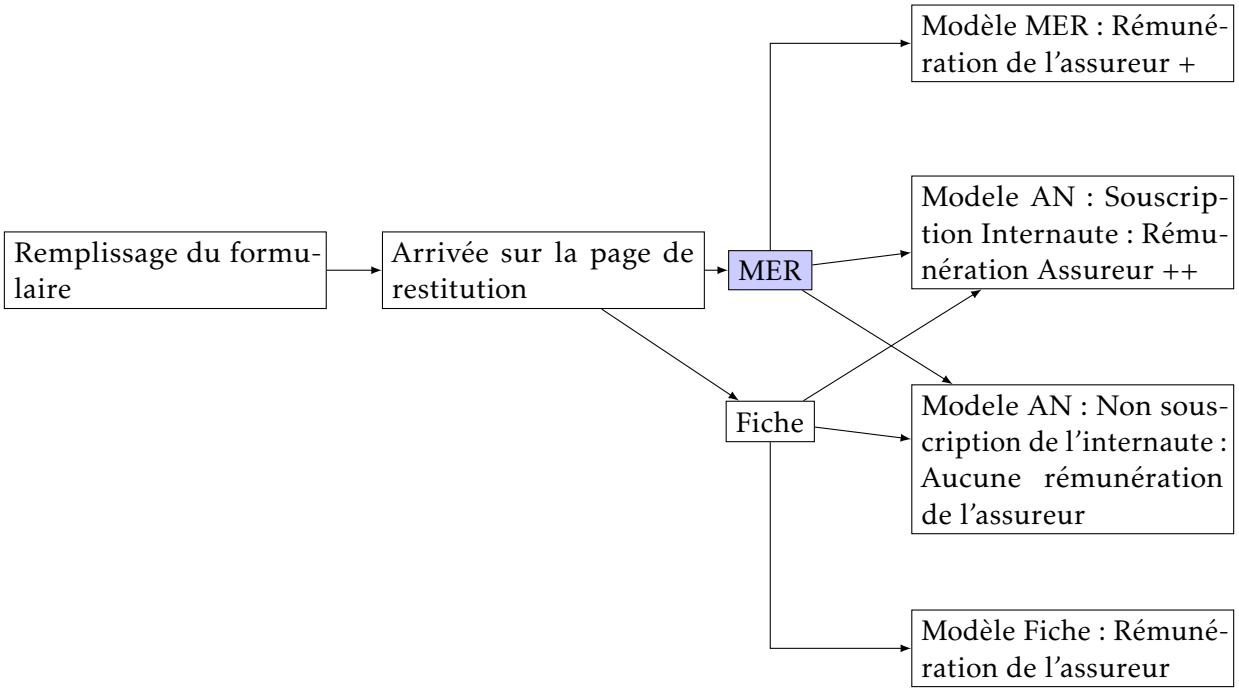


FIGURE 1.3 – Exemple de parcours internaute

et l'internaute est réalisée. Le comparateur sera donc rémunéré par l'assureur, cependant cette rémunération varie en fonction de la qualification d'un internaute. Plus l'internaute est qualifié, c'est à dire intéressé, plus la rémunération est élevée. Les deux types de mise en relations (Fiche et MER) permettent déjà une qualification de l'internaute. Cependant, une dernière étape peut s'ajouter à ce processus en fonction du modèle économique défini entre le comparateur et l'assureur. Cette étape est l'étape de *Souscription*.

Souscription L'internaute mis en relation a finalement souscrit un contrat d'assurance chez cet assureur. Il est alors défini comme *très qualifié*.

Trois principaux modèles économiques sont alors distingués :

Fiche : L'assureur rémunère le comparateur selon le volume de fiches envoyées.

MER : L'assureur rémunère le comparateur selon le volume de MER envoyées

AN : L'assureur rémunère le comparateur selon le volume d'internautes ayant finalement souscrit chez lui.

La figure 1.3 montre le processus de comparaison. Le modèle **AN** est principalement utilisé avec les assureurs proposant de l'assurance automobile. La valeur d'une souscription est plus élevée que la valeur d'une MER ou d'une fiche. Cependant, comme le montre la figure 1.3, si finalement l'internaute ne souscrit pas, le comparateur n'est pas

rémunéré. Or, actuellement, le taux de souscription final pour les internautes envoyés au préalable en fiche est d'environ 4% et est de 12% pour les internautes envoyés en MER. L'objectif est donc de cibler les internautes pour augmenter le nombre d'internautes réalisant des MER et surtout souscrivant finalement chez l'assureur. Le taux de transformation pour les internautes étant envoyés au préalable en fiche étant très bas cela amène à travailler avec des échantillons très petits et surtout déséquilibrés lorsque l'objectif est de prédire les internautes susceptibles de souscrire chez un assureur.

Un comparateur d'assurance en ligne évolue donc dans deux environnements distincts, celui du web et de l'assurance. De plus, le comparateur doit répondre aux attentes des internautes mais également à celles des assureurs partenaires. Cela amène diverses spécificités au domaine de la comparaison d'assurances qui sont détaillées dans la section suivante.

1.1.4 Spécificités du domaine

Pour faire une comparaison d'assurances en ligne, un internaute doit remplir un formulaire contenant diverses questions. Les données nécessaires à la réalisation des différentes études statistiques et notamment à la réalisation de modèles prédictifs proviennent de ces formulaires en lignes. La principale spécificité d'un comparateur d'assurances est que ces formulaires évoluent et sont régulièrement adaptés pour répondre aux contraintes liées au domaine du web et de l'assurance.

Changements réguliers des formulaires

Les formulaires en ligne sont des éléments clés de la comparaison d'assurances. En effet, ils permettent à la société d'afficher aux internautes les tarifs réellement proposés par les assureurs partenaires et non une estimation. D'autre part, ils permettent à la société MeilleureAssurance.com de collecter de nombreuses données pour les différentes analyses statistiques. Cependant, les deux domaines dans lesquels évolue le comparateur impliquent que ces formulaires soient régulièrement modifiés. Le domaine de l'assurance implique différentes contraintes sur les formulaires proposés aux internautes dont l'une des principales vient du fonctionnement entre le comparateur et les assureurs partenaires. D'autre part, le contexte de l'assurance, notamment la législation, peut également contraindre le comparateur à modifier ses formulaires. Concernant le domaine du web ce sont plus particulièrement les internautes qui contraignent le comparateur à modifier régulièrement ses formulaires.

Les systèmes de tarification des assureurs partenaires Un tarif pour une offre d'assurance est unique et répond à un besoin et un profil précis. Généralement, les assureurs utilisent des algorithmes dits de *tarification*, prenant en compte diverses informations afin de proposer un tarif au potentiel client. Les formulaires en ligne du comparateur d'assurances reprennent les informations requises par ces algorithmes pour afficher aux internautes des tarifs correspondant aux tarifs réellement proposés par les assureurs. Cependant, chaque assureur a son propre algorithme de tarification avec ses propres champs de réponses attendus. Les systèmes de tarifications ne sont donc pas homogènes pour tous les assureurs et l'intégration d'un nouvel assureur peut amener le comparateur à adapter son formulaire à l'algorithme du nouvel assureur partenaire. Ces modifications peuvent être des ajouts de questions ou des modifications des descripteurs de questions existantes. Le comparateur étant en pleine croissance, ces modifications dues aux ajouts de partenaires sont fréquentes.

Problématiques liées à la législation La législation peut également amener le comparateur à modifier ses formulaires. Par exemple, la loi Hamon [1], entrée en vigueur le 1er janvier 2015, permet aux consommateurs de changer d'assurance à tout moment après un an de souscription à un contrat d'assurance automobile ou habitation. Un autre exemple est la loi ANI [2](Accord National Interprofessionnel) qui impose à toute entreprise de fournir à ses salariés une complémentaire santé collective, depuis le 1er janvier 2016. Suite à ces lois, les formulaires ont été adaptés afin de les prendre en compte et notamment identifier les profils concernés.

Performances web L'objectif d'un comparateur d'assurances, comme tout site Internet est d'attirer un maximum d'internautes. Cependant, pour un comparateur d'assurances, l'objectif est de permettre aux internautes de comparer différentes offres d'assurances. Afin d'afficher aux internautes les différents tarifs des offres proposées, il est nécessaire que les internautes terminent le remplissage du formulaire. Pour éviter l'abandon des internautes avant la fin des formulaires, ceux-ci doivent être adaptés aux internautes. C'est-à-dire qu'il faut que les formulaires soient compréhensibles et faciles d'utilisation pour l'ensemble des internautes venant sur le comparateur. De plus, les formulaires doivent être ergonomiques et rapides à remplir pour ne pas lasser l'internaute. Pour cela, la société repense régulièrement ses formulaires et ceux-ci sont alors modifiés.

1. https ://www.gouvernement.fr/action/la-loi-consommation
2. https ://www.economie.gouv.fr/entreprises/mutuelle-entreprise-obligatoire

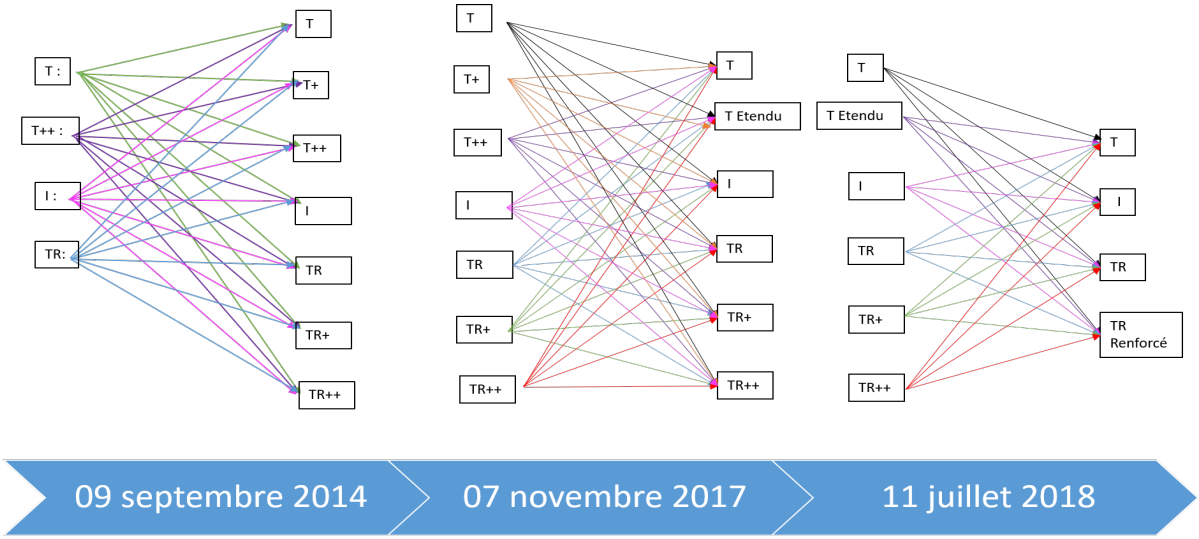


FIGURE 1.4 – Évolution variable "*Niveau de garantie souhaité*"

La Figure 1.4 illustre ces changements de descripteurs de variables. Cette figure représente la variable *Niveau de garantie souhaité* qui indique le niveau de garantie souhaité par l'internaute lors de sa recherche pour une assurance automobile. Dans un modèle de classification, cette variable est l'une des plus discriminantes et son changement impacte fortement le modèle. Avec cette figure, on constate que cette variable a été modifiée à trois reprises aux cours des quatre dernière années. On constate également que cette variable a été modifiée à deux reprises en moins d'un an. Ce qui implique que les différentes analyses utilisant cette variable ont toutes dû être recommencées car les descripteurs sont différents à chaque modification, que ce soit en terme de nombre de modalités ou de sémantique. Or, cette variable est utilisée dans de nombreuses analyses et pour de nombreux outils. Dans le premier cas, cette variable a été modifiée afin de proposer aux internautes des choix plus proches de leurs attentes. Dans les deux dernier cas, cette variable a été modifiée pour répondre aux besoins des assureurs et des offres qu'ils étaient en mesure de réellement proposer aux internautes.

Il est à noter que, dans le même temps, d'autres variables ont été modifiées. Notamment pour l'assurance automobile où sur une période de deux mois, tris questions ont été modifiées et deux questions ont été ajoutées. Le changement des formulaires et en particulier des descripteurs de variables est donc récurrent. Cependant ce n'est pas l'unique spécificité d'un comparateur d'assurances comparé à d'autres sites web.

Historique de navigation

La société *MeilleureAssurance.com* est indépendante de tout organisme de banque ou d'assurance. Une autre spécificité impliquée par ce contexte est l'absence d'historique de navigation des internautes. De nombreux autres domaines du web proposent de la recommandation. Ceux-ci se basent alors sur des méthodes qui utilisent l'historique de navigation des internautes telles que les méthodes de filtrages collaboratifs [87]. Contrairement à de nombreux sites e-commerce, il est difficile de proposer un système de fidélisation ou de récurrence d'achat pour de la comparaison d'assurances. En effet, le délai pour changer de contrat d'assurance est d'un an minimum pour les assurances de consommations. Cela implique que lorsqu'un internaute compare des offres d'assurances, il ne revient au minimum qu'un an après pour de nouveau changer de contrat. Cela rend difficile l'obtention et l'exploitation d'un historique de navigation.

1.2 Problématique et objectif de la thèse

La plupart des analyses statistiques classiques reposent sur des hypothèses fortes : les données proviennent d'une même distribution, les échantillons sont assez grands pour être représentatifs. Dans le cadre de la comparaison d'assurances, ces hypothèses sont régulièrement violées. En effet, les données utilisées pour les analyses statistiques relatives à ce cadre proviennent de systèmes dynamiques où les descripteurs de données sont amenés à changer (évoluer) régulièrement, comme expliqué dans la section 1.1. A chaque changement, la collecte d'un nouvel échantillon est nécessaire et les différentes analyses sont à recommencer. Cela implique d'attendre un certain temps avant d'avoir de nouveau un échantillon conséquent ou parfois de travailler avec de petits échantillons. En effet, même si la population au sens statistique ne change pas, la façon de récolter les données a été modifiée. La durée de vie des différentes analyses est donc limitée. L'objectif de ce travail est alors d'établir un lien entre les données avant la(les) modification(s) des descripteurs de données et après la(les) modification(s) des descripteurs de données. Supposant que les populations, au sens statistique, restent les mêmes indépendamment de la modification des descripteurs de variables, l'objectif final étant de construire un nouvel échantillon représentatif et stable afin de réaliser des analyses statistiques classiques (descriptives, prédictives, ...). Cet appariement entre les données avant et après la modification d'un descripteur devra également répondre aux divers objectifs de la société MeilleureAssurance.com.

Prédiction L'objectif initial de l'entreprise *MeilleureAssurance.com* est la réalisation

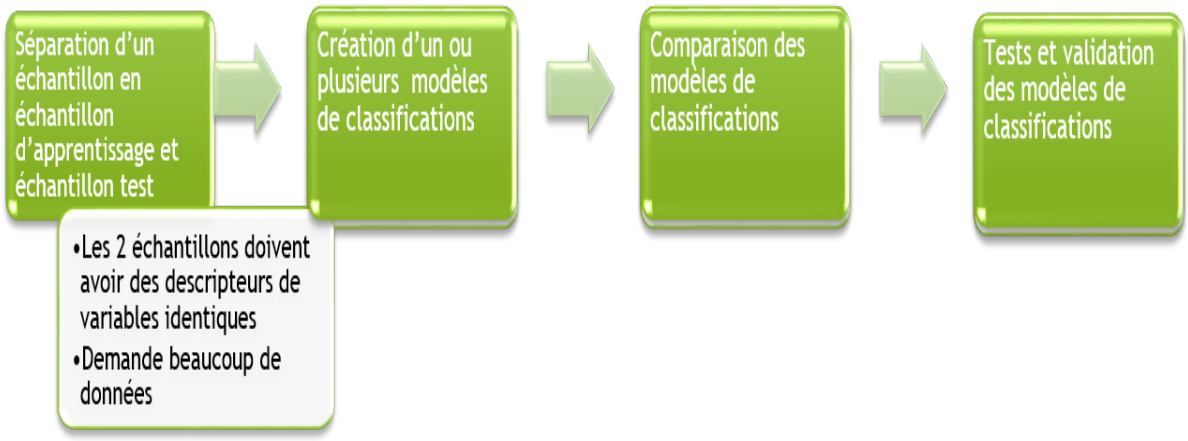


FIGURE 1.5 – Classification classique

d'un modèle prédictif prenant en compte le profil et les attentes de l'internaute afin d'améliorer la pertinence des offres proposées. Ce modèle doit prendre en compte les spécificités liées à la comparaison d'assurances, notamment les changements des formulaires en ligne. Il doit alors être robuste et flexible lors de modifications des descripteurs d'une variable. La figure 1.5 montre le processus standard pour réaliser un modèle de prédiction. Pour ce processus, les échantillons d'apprentissage, de test et de validation sont supposés de structures identiques. C'est à dire, avoir des variables et descripteurs de variables identiques. L'application du modèle repose également sur cette hypothèse. Ce processus peut prendre un certain temps afin d'avoir le modèle performant. Or, lorsque les descripteurs d'une variable sont modifiés, tout ce processus est à recommencer. De plus, le temps de collecte des nouvelles données s'ajoute également à ce processus. Si l'on reprend l'exemple de la figure 1.5, l'objectif est alors d'y inclure une étape transitoire, appariant les données avant et après la modification des descripteurs tel que le montre la figure 1.6. Cette étape transitoire permet alors l'utilisation du modèle initialement créé avec les nouvelles données.

Interprétation L'appariement des données avant et après la modification des descripteurs de variables doit être interprétable. En effet, lorsqu'une variable est modifiée, la société est incapable de connaître réellement l'impact que ce changement a pu avoir sur le comportement des internautes. L'idée est alors d'avoir un système qui permette à la société de comprendre l'impact d'un changement sur le comportement des internautes. Ainsi la société pourra savoir si le comportement des internautes correspond réellement à celui attendu par la société ou

FIGURE 1.6 – Classification chez MA

si l'expérience utilisateur est détériorée. Cette compréhension serait également une aide à la décision dans le choix des modifications puisqu'elle permettrait de juger statistiquement leurs impacts. Si l'on reprend l'exemple donné par la figure 1.4, l'objectif est de pouvoir répondre aux questions de type : *Quel serait le choix des internautes venu avant le 9 septembre 2014, si il revenait le 12 septembre 2014?*, ou *Sur la période du 8 novembre 2017 au 10 juillet 2018, qu'auraient choisi les internautes venus avant le 7 novembre 2017 et qui avaient du Tiers+?*

Généralisation La modification des descripteurs de variables impacte également les différentes analyses statistiques possibles. En effet, la société utilise également les données collectées via les formulaires pour réaliser diverses études marketing, des infographies ou des typologies internautes. Pour pouvoir réaliser ces études, il est nécessaire d'avoir de grands échantillons de données afin qu'ils soient représentatifs. Il arrive fréquemment que ces études soient réalisées selon des profils précis, par exemple, selon un département ou une ville. Pour ces études, il est nécessaire de prendre une période de temps très large pour avoir des échantillons représentatifs. Or, lorsqu'une variable est modifiée, cette période est restreinte à la dernière modification réalisée afin d'avoir une interprétation fiable du choix de l'internaute. La taille de l'échantillon est donc limitée et il est fréquent qu'une étude ne puisse pas être réalisée car l'échantillon est trop faible. De même que pour la classification, la reconstruction d'un échantillon muni des données avant et après la modification permettrait la réalisation de nouvelles études. D'autre part, lors d'une modification, les études réalisées avant la modification deviennent rapidement obsolètes. L'appariement entre les données permettrait d'éviter l'obsolescence prématurée de ces études.

Contrairement à la plupart des sites e-commerce, lorsqu'un internaute vient comparer un produit d'assurance, il est rare qu'il revienne sur le site. A la problématique liée aux changements des descripteurs de variables s'ajoute un problème alors de données

IDInternaute	Xi	Yi
1	x1	y1
2	x2	y2
3	x3	y3

(a) Données complètes

IDInternaute	Xi	Yi
1	x1	
2	x2	
3		y3

(b) Données observées

FIGURE 1.7 – Exemple de données

manquantes. En effet, le choix d'un internaute n'est disponible que pour la variable avant ou après la modification mais jamais les deux. Or une autre hypothèse forte sur laquelle repose la plupart des analyses statistiques est que les échantillons utilisés ne comportent pas de données manquantes. La figure 1.7a illustre les données disponibles. On constate alors que pour chaque couple, seule une donnée partielle est observée. Cette propriété peut être formalisée de la façon suivante : de manière générale, un questionnaire dit « Questionnaire dynamique » correspond à un couple de questionnaires (respectivement $Q_{\mathbf{x}}$ et $Q_{\mathbf{y}}$), où les deux sont remplis par le même nombre n d'individus pour une modalité (l'ensemble des résultats sont respectivement les variables $\boldsymbol{x}$ et $\boldsymbol{y}$). Ces variables ont la même signification pour les deux questionnaires mais pas nécessairement le même nombre de modalités (p et q respectivement).

Période avant la redéfinition du site web n^- internautes ont renseigné la variable $\boldsymbol{x}$, produisant des réalisations *observées* $\mathbf{x}^- = (\boldsymbol{x}_1^-, \ldots, \boldsymbol{x}_{n^-}^-)$. Comme on vient de le préciser, la variable $\boldsymbol{y}$ n'a jamais été renseignée par contre, produisant des réalisations *non observées* $\mathbf{y}^- = (\boldsymbol{y}_1^-, \ldots, \boldsymbol{y}_{n^-}^-)$.

Période après la redéfinition du site web De façon symétrique, n^+ internautes ont renseigné la variable $\boldsymbol{y}$, produisant des réalisations *observées* $\mathbf{y}^+ = (\boldsymbol{y}_1^+, \ldots, \boldsymbol{y}_{n^+}^+)$. Comme attendu, la variable $\boldsymbol{x}$ n'a jamais été renseignée par contre, produisant des réalisations *non observées* $\mathbf{x}^+ = (\boldsymbol{x}_1^+, \ldots, \boldsymbol{x}_{n^+}^+)$.

De plus, on considère que les individus précédents peuvent être séparés en deux groupes de tailles respectives n^- et n^+, avec $n = n^- + n^+$. Nous avons donc une partition $\mathbf{x} = (\mathbf{x}^-, \mathbf{x}^+)$ et $\mathbf{y} = (\mathbf{y}^-, \mathbf{y}^+)$, qui seront notés simplement $\mathbf{x}^-, \mathbf{x}^+, \mathbf{y}^-$ et $\mathbf{y}^+$. Le problème étudié dans le cadre de ce travail contient de nombreuses données manquantes rendant le choix des internautes après la modification de la variable incertain. Le cadre probabiliste permettant la modélisation d'expériences aléatoires telles que celles présentes dans le

cadre de cette étude est alors particulièrement adapté. Dans ce travail, deux expériences aléatoires peuvent être définies. La première, avant la modification de la variable et la seconde après la modification de la variable. Ces deux expériences se caractérisent par des univers, des attributs et des distributions différentes. En effet, la première expérience $\mathbf{x}$ est caractérisée par un univers fini de taille p tel que $\Omega_x = \{x_{j1}, \ldots, x_{jp}\}$ alors que la seconde $\mathbf{y}$ est caractérisée par un univers fini de taille q tel que $\Omega_y = \{y_{j1}, \ldots, y_{jq}\}$, où j est l'indice de la variable modifiée. Pour chacune des deux expériences aléatoires, les évènements élémentaires sont définis par les descripteurs de chacune des variables. La modification des descripteurs d'une variable implique alors de travailler dans des espaces différents et avec des données manquantes. La réalisation d'un modèle prédictif permettant une meilleure pertinence des offres proposées aux internautes devient alors complexe.

1.3 Conclusion

Dans ce premier chapitre, nous avons présenté la société MeilleureAssurance.com, qui est un comparateur d'assurances en ligne, et sa problématique concernant la réalisation d'un modèle prédictif afin de proposer aux internautes les offres les plus pertinentes. Dans un second temps, les spécificités liées au domaine de l'assurance et du web ont également été abordées. Ces spécificités impliquent différents verrous scientifiques rendant complexe la réalisation du modèle avec les méthodes et outils existants. En effet, une des spécificités d'un comparateur d'assurances est le changement régulier des descripteurs des variables des formulaires. L'objectif principal de la société MeilleureAssurance.com devient la réalisation d'un modèle prédictif robuste et flexible lors de modifications de données. Dans le chapitre suivant, nous détaillons les différentes approches de classifications existantes. Ces approches incluent les méthodes de classification usuelles pouvant être utilisées pour la création du modèle prédictif final, les méthodes de classification évolutives plus particulières au domaine du web et les méthodes de Transfer Learning pouvant répondre à la problématique du changement d'espace des variables.

Chapitre 2

Etat de l'art en classification et transfert de connaissances

> Le seul moyen de faire une méthode instructive et naturelle, est de mettre ensemble des choses qui se ressemblent et de séparer celles qui diffèrent les unes des autres
>
> *Buffon, 1749*

La spécificité des données d'un comparateur d'assurances, impose une certaine complexité pour réaliser les différentes analyses statistiques et particulièrement l'apprentissage de modèles de classification. Dans ce chapitre, nous présenterons les différentes approches de classification existantes. Dans une première section, nous introduirons la classification "classique", présenterons ses principales méthodes, et indiquerons pourquoi elles ne peuvent répondre à la problématique. Dans une seconde section, nous présenterons les méthodes de classifications évolutives existantes, souvent utilisées dans le domaine du web. Enfin, dans une troisième section nous introduirons le transfert de connaissances, qui se place dans le cadre de l'apprentissage automatique, et présenterons également les principales méthodes se plaçant dans notre cadre.

2.1 Classification supervisée et non supervisée

La classification désigne un ensemble de méthodes permettant de classer/regrouper un ensemble d'observations en classes homogènes à partir de différentes caractéristiques. Au fil du temps, différentes approches ont été proposées. On peut distinguer les premières approches (algorithmiques, géométriques, heuristiques) qui se basaient essentiellement sur la dissimilarité des objets à classer telle que la célèbre méthode des K-means [74], de l'approche statistique [85], plus récente, qui se base sur des modèles

probabilistes pour formaliser l'idée de classes [22]. Avec l'apparition de nouveaux domaines d'applications, notamment liés au web, de nouvelles approches sont également apparues. En effet, les applications du web telles que le web-mining entraînent des modèles qui doivent devenir évolutifs car les profils d'individus évoluent dans le même temps. Pour traiter cette approche, des méthodes de classification évolutives ont alors été créées. Une autre approche est le transfert de connaissances, qui permet d'appliquer un même modèle sur des données différentes. Ces applications concernent des domaines variés mais touchent particulièrement le domaine du web, notamment pour la classification de textes et/ou d'images.

2.1.1 Formalisation de la classification

Dans un problème de classification, une variable réponse $z \in \mathcal{Z}$ doit être prédite venant d'un jeu de d variables caractéristiques $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d)$ à valeur dans un espace mesurable $\mathcal{X}$. Dans notre jeu de données l'espace $\mathcal{X}$ est mixte et l'espace $\mathcal{Z}$ est catégoriel. Dans ce travail, nous disposons d'un ensemble de n observations à classer. Ces n observations, décrites par d les variables caractéristiques sont notées $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ et la variable réponse est décrite par $\mathbf{z} = (z_1, \ldots, z_n)$, où les n observations sont supposées être des réalisations indépendantes et identiquement distribuées. La variable $z \in \{1, \ldots, g\}$, introduit l'appartenance à une des g classes telle que $z = k$ si $\boldsymbol{x}$ appartient à la k_{eme} classe. L'espace d'appartenance de la classe sera notée $\mathcal{Z} = \{1, \ldots, g\}$. Dans un contexte statistique, les couples $(\boldsymbol{x}, z)$ sont supposés être des réalisations du vecteur aléatoire $(\boldsymbol{X}, \boldsymbol{Z})$ où $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_d)$. L'objectif de la classification est alors d'associer le vecteur $\boldsymbol{x}$ à une des classes g et donc d'établir une règle de décision ϕ tel que le vecteur $\boldsymbol{Z} \in \{1, \ldots, g\}$ soit associé au vecteur $\boldsymbol{X} \in \mathcal{X}$:

$$\phi : \mathcal{X} \to \mathcal{Z}. \tag{2.1}$$

Dans le cadre probabiliste, la règle de décision est la règle minimisant le risque conditionnel, tel que :

$$R(\phi|\boldsymbol{x}) = 1 - P(\boldsymbol{Z} = \phi(\boldsymbol{x})|\boldsymbol{X} = \boldsymbol{x}). \tag{2.2}$$

La règle de décision optimale ϕ^*, appelée aussi **règle de Bayes** [48], minimise le risque d'erreur conditionnel $R(\phi|\boldsymbol{x})$ pour chaque observation. Cette règle consiste donc à affecter l'observation $\boldsymbol{x}$ à la classe la plus probable a-posteriori :

$$\phi^*(\boldsymbol{x}) = \underset{k=1,\ldots,g}{\operatorname{argmax}} P(\boldsymbol{Z} = k|\boldsymbol{X} = \boldsymbol{x}). \tag{2.3}$$

Classiquement, la règle de décision est construite à partir d'un échantillon dit "d'apprentissage". Selon les données dont on dispose dans l'échantillon d'apprentissage, on distingue deux types de classification. Si dans l'échantillon d'apprentissage on ne dispose que des valeurs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ prises par les d variables explicatives, on parlera alors de **classification non supervisée**. Au contraire, si on dispose des valeurs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ prises par les d variables explicatives et de leur appartenance $z_1, \ldots, z_n$ aux g classes, on parlera alors de **classification supervisée**. A noter, en classification, la variable $\boldsymbol{Z}$ est discrète. Lorsque la variable $\boldsymbol{Z}$ est continue, *i.e* $\boldsymbol{Z} \in [0,1]$ ou $\mathbb{R}$, on parle de régression.

Notre objectif étant de faire de la prédiction, la classification non supervisée ne sera pas détaillée dans ce travail. Dans la suite de ce chapitre, nous détaillons la classification supervisée et ses différentes approches.

2.1.2 Classification supervisée

En classification supervisée, aussi appelée analyse discriminante [49], [111], l'objectif est de construire une règle de décision (aussi appelée parfois classifieur) à partir d'observations où la classe à laquelle elles appartiennent est connue. C'est ce qui la distingue de la classification non-supervisée où les classes sont inconnues. La classification supervisée peut avoir pour objectif de trouver une représentation qui permette une interprétation des groupes grâce aux variables explicatives. L'objectif, dans ce cas, est donc descriptif. D'autre part, l'objectif peut être de définir la meilleure affectation d'un nouvel individu où seules les valeurs des variables explicatives sont connues. Dans ce cas, l'objectif est décisionnel (prédictif). Dans ce mémoire, nous nous intéressons plus particulièrement à l'aspect décisionnel qui correspond au cadre de la thèse.

Principales étapes

Le problème de la classification supervisée peut être défini comme la prédiction d'une observation $\boldsymbol{x}$, décrite par d variables explicatives, à une classe k parmi g classes définies a priori. La prédiction d'une observation $\boldsymbol{x}$ se fait à l'aide d'un échantillon d'apprentissage défini par

$$A = \{(\boldsymbol{x}_i, z_i) : \boldsymbol{x}_i \in \mathcal{X}, z_i \in \{1, \ldots, g\}, i = 1, \ldots, n\} \tag{2.4}$$

où le vecteur $\boldsymbol{x}_i$ correspond à la valeur prise par l'observation i pour les d variables explicatives et z_i correspond à la classe à laquelle il appartient. A partir de cet échantillon la règle de Bayes $\phi(\boldsymbol{x})$ peut être créée. En statistique, pour estimer la règle de Bayes, il

existe deux approches.

L'approche générative modélise la distribution jointe des données $(\boldsymbol{Z},\boldsymbol{X})$, pour en déduire ensuite la distribution conditionnelle aux classes $(\boldsymbol{X}|\boldsymbol{Z})$ par l'application d'une règle de Bayes. Autrement dit, l'approche générative estime la densité conditionnelle aux classes $P(\boldsymbol{x}|\boldsymbol{Z}=k)$, par l'intermédiaire de la loi jointe $P(\boldsymbol{x},\boldsymbol{Z}=k)$, $k=1,\ldots,g$.

L'approche prédictive repose sur l'estimation des probabilités a posteriori $P(\boldsymbol{Z}=k|\boldsymbol{x})$, $k=1,\ldots,g$.

Nous présentons désormais, les méthodes usuelles relatives aux deux approches que l'on pourra retrouver dans les livres [48] et [40].

Méthodes génératives

Dans l'approche générative, deux types de méthodes peuvent être utilisées. Selon les hypothèses effectuées sur les densités des covariables [111] on distingue les méthodes paramétriques telles que l'analyse discriminante, des méthodes non-paramétriques telles que la méthode du noyau. Il existe de nombreuses méthodes génératives paramétriques et non-paramétriques. Dans ce chapitre, nous présenterons les plus répandues de chacune des deux catégories, soit l'analyse discriminante linéaire et la méthode du noyau.

Analyse discriminante linéaire & quadratique [37], [48] L'analyse discriminante linéaire est considérée comme une référence pour les méthodes de classification car elle a de nombreuses propriétés statistiques et fournit de bonnes performances. C'est une méthode générative paramétrique, qui suppose donc que les densités des covariables suivent des lois paramétriques. L'analyse discriminante linéaire émet l'hypothèse que les densités conditionnelles aux classes $P(\boldsymbol{x}|\boldsymbol{Z}=k)$, $k=1,\ldots,g$ sont des lois gaussiennes $\mathcal{N}(\boldsymbol{\mu}_k,\Sigma)$ de moyenne $\boldsymbol{\mu}_k$ et d'une matrice de covariance commune Σ dont le but est de trouver une séparation linéaire entre les classes :

$$P(\boldsymbol{x}|\boldsymbol{Z}=k)=\frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)'\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right). \tag{2.5}$$

Dans le cas $g=2$, avec un coût d'erreur équiprobable, la règle de Bayes s'écrit :

$$\phi^*(\boldsymbol{x})=1 \iff (\boldsymbol{x}-\frac{\boldsymbol{\mu}_1+\boldsymbol{\mu}_2}{2})'\Sigma^{-1}(\boldsymbol{\mu}_1+\boldsymbol{\mu}_2)\geq 0.$$

Les paramètres des distributions sont ensuite estimés par leurs valeurs empiriques.

$$\hat{\boldsymbol{\mu}}_k = \bar{\boldsymbol{x}}_k, k = 1, \ldots, g$$

et

$$\hat{\Sigma} = \frac{\sum_{k=1}^{g} \sum_{i/y_i=k} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)'}{n - g}.$$

Lorsque l'on suppose la matrice de covariance libre pour chaque classe, on se place alors dans le cadre de l'analyse discriminante quadratique. Dans ce cas l'analyse discriminante amènera à une séparation quadratique des classes, et pour chaque classe il faudra estimer la matrice de covariance Σ_k, $k = 1, \ldots, g$ tel que :

$$\hat{\Sigma}_k = \frac{\sum_{i/y_i=k} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)'}{n_k - g}$$

où n_k est le nombre d'observations appartenant à la classe k.

Méthode du noyau Contrairement à la méthode précédente, ici, on se place dans un cadre non paramétrique. Il n'y a donc pas d'hypothèse sur la loi des densités de chacune des classes. Pour estimer la densité conditionnelle aux classes $P(\boldsymbol{x}|\boldsymbol{Z} = k)$, $k = 1, \ldots, g$ on utilise la méthode du noyau. C'est une des méthodes usuelles non paramétriques. Les densités sont estimées par :

$$P(\boldsymbol{x}|\boldsymbol{Z} = k) = \frac{1}{n_k h^d} \sum_{i/z_i=k} K\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}\right),$$

où K est une densité de probabilité (Gausienne, uniforme, etc.) et h le paramètre de lissage. Dans cette méthode, le choix du paramètre de lissage h est important. En effet, s'il est trop petit, la densité estimée aura trop de modes. A l'inverse, s'il est trop grand, la densité estimée sera trop lisse. Plusieurs stratégies ont été proposées pour choisir ce paramètre. Une des méthodes est un choix par validation croisée sur l'échantillon d'apprentissage. Comme c'est une méthode non-paramétrique, l'échantillon d'apprentissage doit être grand pour qu'elle soit robuste. De plus, elle peut vite s'avérer coûteuse en temps, si h est grand.

Le principal reproche qui est fait à ces méthodes est de ne pas assez prendre en compte l'objectif de prédiction dans l'apprentissage de la règle de classement. De plus l'analyse discriminante linéaire s'applique avec des variables caractéristiques quantitatives, alors que nos variables caractéristiques peuvent être mixtes. Nous allons maintenant présenter les méthodes prédictives, qui contrairement aux méthodes génératives

ont l'avantage de prendre directement en compte l'objectif de prédiction.

Méthodes prédictives

Les méthodes prédictives prennent directement en compte l'objectif de prédiction. Cependant elles ne font pas d'hypothèses sur la distribution des covariables. Parmi les méthodes prédictives, on peut distinguer :

- les méthodes non paramétriques (K plus proches voisins [28], deep learning [31])
- les méthodes à base d'arbre(CART) [20]
- les méthodes semi-paramétriques (régression logistique) [4]
- les méthodes de recherche d'un hyperplan optimal comme les SVM (support vecteur machine) [112]

Dans ce travail, nous présenterons, la régression logistique, l'approche à base d'arbre et les SVM.

Régression Logistique C'est une approche prédictive semi-paramétrique. Contrairement à la méthode LDA où on fait une hypothèse sur chacune des distributions conditionnelles aux classes, cette méthode suppose que la différence entre les logarithmes des densités des classes est linéaire par rapport à $\boldsymbol{x}$. C'est cette hypothèse qui fait que l'on parle d'approche semi-paramétrique et non paramétrique. Si on suppose que $g = 2$ et qu'on pose $\pi(\boldsymbol{x}) = P(\boldsymbol{Z} = 1|\boldsymbol{x})$, l'équation de la régression logistique s'écrit :

$$logit(\pi(\boldsymbol{x})) = ln(\frac{\pi(\boldsymbol{x})}{1-\pi(\boldsymbol{x})}) = \beta_0 + \sum_{j=1}^{d} \beta_j \boldsymbol{x}_j$$

où $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_d)$ sont des coefficients réels qui seront définis lors de l'apprentissage du modèle et $\boldsymbol{x}$ les variables caractéristiques du modèle associé aux coefficients $\boldsymbol{\beta}$.

Ce modèle peut également s'écrire :

$$\pi(\boldsymbol{x}; \boldsymbol{\beta}) = \frac{\exp(\beta_0 + \sum_{j=1}^{d} \beta_j \boldsymbol{x}_j)}{1 + \exp(\beta_0 + \sum_{j=1}^{d} \beta_j \boldsymbol{x}_j)}.$$

En général, les paramètres $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_d)$ sont estimés par maximisation de la

vraisemblance conditionnelle pour l'échantillon d'apprentissage A, défini par :

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \ln P(z_i|\boldsymbol{x}_i)$$

d'où

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} (z_i \boldsymbol{\beta}' \boldsymbol{x}_i - \log(1 + \exp \boldsymbol{\beta}' \boldsymbol{x}_i)).$$

La maximisation de cette vraisemblance se fait en dérivant par rapport aux β. On obtient alors

$$\sum_{i=1}^{n} (z_i - \pi(\boldsymbol{x}_i; \boldsymbol{\beta}))\boldsymbol{x}_i = 0$$

qui n'est pas une équation linéaire en $\boldsymbol{\beta}$. Cependant, elle peut être résolue à l'aide d'un algorithme de Newton-Raphson .

Une fois les $\boldsymbol{\beta}$ estimés, il est possible d'utiliser la règle du *Maximum a posteriori* (MAP) comme règle de décision.

Généralisation Le modèle qui vient d'être présenté dans le cas binaire peut se généraliser au cas d'une variable G avec $g > 2$ modalités. On définit $\pi(\boldsymbol{x}) = P(\boldsymbol{Z} = k|\boldsymbol{x})$. On fixe alors une modalité de référence, par exemple g, et on réalise $g-1$ régressions logistiques de k versus g, $k = 1, \ldots, g-1$:

$$\ln\left(\frac{\pi_k(\boldsymbol{x})}{\pi_g(\boldsymbol{x})}\right) = \beta_{0k} + \sum_{j=1}^{d} \beta_{jk}\boldsymbol{x}_j \quad \forall k = 1, \ldots, g-1.$$

Cette procédure ne dépend pas de la classe de référence et on peut facilement en déduire :

$$P(\boldsymbol{Z} = k|\boldsymbol{x}) = \frac{\exp(\beta_{0k} + \ldots + \beta_{dk}\boldsymbol{x}_d)}{1 + \sum_{l=1}^{g-1} \exp(\beta_{0l} + \ldots + \beta_{dl}\boldsymbol{x}_d)}, k = 1, \ldots, g-1.$$

La régression logistique est particulièrement appréciée dans de nombreux domaines (médecine, finance, assurance, marketing, ...) car c'est un modèle facile d'interprétation, notamment avec les *odds-ratio*, qui peuvent être interprétés comme le risque multiplicatif. Un *odds* est défini par :

$$odds(\boldsymbol{x}) = \frac{\pi(\boldsymbol{x})}{1 - \pi(\boldsymbol{x})}$$

qui représente le risque d'appartenance à la classe $k = 1$ par rapport à la classe $k = 2$ lorsque $\boldsymbol{X} = \boldsymbol{x}$. On définit alors de la même façon les *odds-ratio* par

$$odds\text{-}ratio(\boldsymbol{x}_j, \boldsymbol{x}_l) = \frac{odds(\boldsymbol{x}_j)}{odds(\boldsymbol{x}_l)}$$

qui représente le risque d'appartenance à la classe $k = 1$ par rapport à la classe $k = 2$ pour un individu ayant la variable $\boldsymbol{x}_j$ par rapport à un individu ayant la variable $\boldsymbol{x}_l$. Par exemple, si l'odds-ratio est de 2, alors on a 2 fois plus de risque d'appartenir à la classe $k = 1$ qu'à la classe $k = 2$ en ayant la variable $\boldsymbol{x}_j$ par rapport à la variable $\boldsymbol{x}_l$

La régression logistique a pour avantage d'être assez générale car elle ne fait pas d'hypothèse sur chacune des classes et demande un faible nombre de paramètres comparée à l'analyse discriminante linéaire.

Arbre de classification Ils relèvent des méthodes prédictives non paramétriques dont le but est de construire un classifieur simple et interprétable. Ces méthodes ont été introduites dans les années 1950 et ont connu une seconde expansion en 1984 avec le logiciel CART (Classification and regression Tree) proposé par Breiman [20]. De nos jours, les arbres de décisions tels que CART ou C4.5 sont devenus des outils standards de data-mining.

D'une manière générale, les arbres de classification (parfois aussi appelés arbres de décision) sont des méthodes qui permettent d'obtenir des modèles à la fois explicatifs et prédictifs. Il s'agit d'une méthode itérative, dite de partitionnement récursif des données [98]. L'idée de la méthode est de construire des classes d'individus, les plus homogènes possibles, en posant une succession de questions binaires (de type oui/non) sur les variables caractéristiques de chaque individu. Finalement, l'arbre de sortie peut être vu comme un diagramme orienté où chaque nœud final est une classe. Plus précisément, un arbre de classification prend en entrée l'échantillon d'apprentissage A. A chaque itération, un nœud intermédiaire est divisé selon une coupure c prenant la forme d'un test pour construire de nouveaux nœuds les plus homogènes possible au sens de la variable à expliquer. Si la variable prédictive $\boldsymbol{x}_j$ est qualitative, la coupure définira si la caractéristique est présente ou non dans la classe prédite. Si la variable prédictive $\boldsymbol{x}_j$ est quantitative, la coupure sera de la forme $\boldsymbol{x}_j > c$?. L'arbre final dit "maximal" est obtenu lorsqu'aucun noeud ne peut plus être divisé. Chaque noeud final est alors affecté à l'une des modalités de la variable à expliquer. Pour trouver

l'appartenance d'une nouvelle observation, il suffit alors de descendre l'arbre en fonction des noeuds.

Pour mesurer l'apport de la classification, il est commun d'utiliser l'entropie définie par Shannon [103]. L'entropie mesure la quantité moyenne d'information apportée par un noeud s, et donc l'impureté d'un noeud. Elle peut être définie comme :

$$Entropie(s) = -\sum_{k=1}^{g} P(k|s) * \log P(k|s)$$

avec $P(k|s)$ représentant la proportion de la classe k dans le noeud s.

Une autre mesure utilisée est le Gain d'information qui mesure la différence d'entropie avant et après le partitionnement d'un noeud s selon le test t. C'est à dire, qu'il mesure l'information gagnée en partitionnant le noeud s selon le test t.

$$Gain(s,t) = Entropie(s) - \sum_{j=1}^{n} (\frac{|s_j|}{s} * Entropie(s_j))$$

où n est le nombre de modalité du test t partitionnant le noeud s. Par exemple, si t est binaire, $n = 2$. Cette mesure peut être utilisée pour classer les tests et construire l'arbre de décision où chaque noeud situe le test ayant le plus haut gain d'information parmi les tests qui ne sont pas encore considérés. La mesure du gain, bien que donnant de bons résultats, a tendance à sélectionner le test ayant des sorties multiples. Ce biais, inhérent au critère du gain, peut être corrigé par la mesure *split-info* qui donne le potentiel d'information généré en divisant le test t en l sous tests.

$$split\text{-}info(s,t) = -\sum_{i=1}^{l} \frac{t_i}{t} * log \frac{t_i}{t}.$$

Où l est le nombre d'arité du test t. Le gain ratio quant à lui donne la proportion utile d'information générée par la répartition créée par le test.

$$Gain\text{-}ratio(s,t) = \frac{Gain(s,t)}{split\text{-}info(s,t)}.$$

C'est la mesure utilisée par Quinlan [88] pour les arbres C4-5 afin de déterminer la meilleure répartition de tests. A chaque noeud, C4-5 choisit le test donnant la meilleure répartition du jeu de données en sous-jeux de données enrichissant

une classe ou une autre. Le test avec le plus haut gain d'information normalisé est choisi pour prendre la décision.

La décision générée, étant donné l'échantillon d'apprentissage A, crée souvent un arbre qui sur-apprend les données et qui est très sensible aux bruits. Pour éviter ce sur-apprentissage on va utiliser la technique de l'élagage. L'élagage est une technique de machine learning qui réduit la taille de l'arbre en supprimant les sections de l'arbre qui ont le moins de pouvoirs de classifications. L'algorithme d'élagage est basé sur une estimation pessimiste du taux d'erreur.

Les arbres de décision ont pour avantage de donner un modèle lisible et de permettre de trouver facilement les variables discriminantes dans un grand échantillon de données. Contrairement à d'autres méthodes de classification (régression logistique, SVM, etc.), les arbres de décision sont très intuitifs et fournissent une représentation graphique, claire et facile à interpréter, d'un processus de classification d'observations.

Support vector machines Les méthodes "machine à vecteur support" ou *support vector machines* (SVM) en anglais, ont été introduites par Vapick [112] au milieu des années 90. Elles appartiennent à la famille des classifieurs binaires très utilisée dans la communauté du machine learning. Les SVM ne font pas d'hypothèses sur la distribution $(\boldsymbol{Z}|\boldsymbol{X})$ et ont pour but de trouver le meilleur hyperplan séparant deux groupes afin d'avoir une séparation linéaire produisant une marge maximale. Les SVM se distinguent par le fait de ne pas contraindre l'espace de recherche à l'espace d'origine. Généralement, la recherche de l'hyperplan est effectuée sur des espaces de grandes dimensions. La règle de décision peut être définie par :

$$M(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i w_i K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) + \beta_0$$

où les α_i et β_0 sont les coefficients des vecteurs supports, $w_i = 1$ si l'observation appartient à la classe $k = 1$ et -1 sinon. K est une fonction dite noyau, définie par $K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) =< h(\boldsymbol{x}_i), h(\boldsymbol{x}_{i'}) >$, où $h(.)$ est l'opérateur de transformation des données. Cette règle permet d'affecter une nouvelle observation x en fonction du signe de la quantité $M(\boldsymbol{x})$. L'estimation des α_i et β_0 correspond à un problème d'optimisation convexe que des outils d'optimisation standards peuvent résoudre. Dans la pratique, un dernier paramètre habituellement noté γ contraint le problème. Celui-ci est réglé de manière empirique par l'utilisateur.

Limitation des méthodes précédentes Ces méthodes que l'on peut qualifier de "classiques" sont largement utilisées en classification. Cependant, elles reposent sur des hypothèses fortes qui les rendent difficilement applicables dans le domaine du web. Ces méthodes supposent que les échantillons sur lesquelles les modèles sont appris, testés et appliqués ne changent pas au cours du temps. Or, pour de nombreuses applications, il est de plus en plus courant que le profil des individus change ou même que les variables évoluent. C'est par exemple, ce qui se passe dans le cadre de notre travail. Dans la partie suivante, nous introduisons la classification dite "évolutive".

2.1.3 Classification évolutive

Historiquement, les méthodes d'apprentissage telles que celles présentées dans la partie précédente font l'hypothèse que toutes les données sont disponibles au même moment et peuvent être chargées en mémoire pour pouvoir réaliser l'apprentissage. Cependant, dans les nouveaux domaines d'application de la fouille de données, telle que la modélisation des utilisateurs au sein d'un réseau social ou le web mining, le nombre de données peut vite exploser. De plus, notamment dans le domaine du web, le profil des internautes peut évoluer dans le temps et le modèle doit être adapté. Les algorithmes classiques statiques doivent alors être adaptés pour répondre à ces nouvelles contraintes. Pour répondre à ces problématiques, les algorithmes d'apprentissage incrémentaux sont utilisés. Si l'on reprend la définition de Polikar [86], [3] l'apprentissage incrémental doit répondre aux critères suivants :

- Être capable d'apprendre des connaissances supplémentaires à partir de nouvelles données
- Ne pas avoir besoin des données d'origine
- Préserver les connaissances déjà acquises
- Pouvoir apprendre de nouvelles classes pouvant être introduites avec de nouvelles données

Fonctionnement général

En informatique, l'apprentissage incrémental correspond aux algorithmes permettant d'entraîner un modèle de manière incrémentale. Dans un algorithme d'apprentissage incrémental, l'échantillon d'apprentissage ne possède pas toutes les données d'apprentissage au même moment mais celles-ci arrivent au fil du temps. L'algorithme apprend alors le modèle au fur et à mesure que ces données arrivent. Il faut bien noter que l'algorithme ne réapprend pas le modèle à partir de 0 et n'a pas besoin de sto-

cker l'ensemble des données du modèle, puisqu'il enrichit le modèle initial sans le réapprendre entièrement. Le modèle doit donc pouvoir être modifié et ses paramètres être ajustés au fur et à mesure des nouvelles observations, néanmoins il ne doit pas oublier la connaissance acquise à partir des exemples précédents. On peut distinguer deux types d'approches pour la classification incrémentale, le **système adaptatif** et le **système évolutif** [17] et [3].

Système adaptatif Dans un algorithme adaptatif, la structure des données initiales ne change pas au cours de l'apprentissage. Les algorithmes d'apprentissage vont apprendre de manière incrémentale les paramètres mais la structure reste fixe. Cet apprentissage des paramètres peut être considéré comme un algorithme « d'adaptation ».

Système évolutif Contrairement au système adaptatif, pour les algorithmes évolutifs, la structure peut être modifiée. C'est à dire que les paramètres sont appris de manière incrémentale, mais que la structure peut aussi être modifiée, comme par exemple lors de l'apparition d'une nouvelle classe.

Le système évolutif est bien plus flexible qu'un système adaptatif car il permet, en cours d'utilisation, d'ajouter des classes supplémentaires lorsque cela s'avère nécessaire.

Plusieurs approches ont été proposées pour réaliser une classification incrémentale. Une partie de ces méthodes consiste à rendre des algorithmes classiques, tels que les SVM ou les arbres, incrémentaux. La seconde approche repose sur les systèmes d'inférence floue. Voici une description de ces approches.

SVM incrémental : Les premières approches des SVM incrémentaux ont été proposées par Syed [105] et Ruping [96]. L'idée est de gérer l'ensemble de vecteurs supports de façon incrémentale. Lorsqu'une nouvelle donnée d'apprentissage est mal classée ou est à l'intérieur de la marge, la séparation est recalculée à partir des vecteurs supports et de la nouvelle observation. D'autres versions incrémentales des SVM ont été proposées, dont celle de Fung et Mangasarian [41] qui proposent un PSVM - Proximal SVM. Dans cette approche, une frontière n'est pas vue comme un plan mais comme un espace (plusieurs plans). L'ensemble de points situé dans l'espace proche, autour de l'hyperplan, et les observations supportant les vecteurs supports sont gardés. Cet ensemble évolue en supprimant les observations trop anciennes et en ajoutant les nouvelles observations. De cette façon le SVM est rendu incrémental. L'algorithme LASVM [71] est une version plus récente. Dans cette version, la sélection des points à intégrer dans la solution est réalisée activement. L'avantage de cette version est que l'apprentissage peut être interrompu

à tout moment, avec une étape de fin facultative (qui correspondrait à supprimer les vecteurs supports devenus obsolètes). Cette étape de fin, faite régulièrement pendant l'apprentissage, permet de rester proche des solutions optimales.

La difficulté pour les SVM incrémentaux est la gestion de nouvelles classes. En effet, si une nouvelle classe apparaît, elle contiendra peu d'individus, donc en maximisant la marge, un SVM aura plutôt tendance à isoler ces quelques individus, au lieu de généraliser la représentation de cette nouvelle classe.

Arbre de classification : Les arbres de références, présentés précédemment (C4.5, CART) ne sont pas incrémentaux. Cependant, des versions incrémentales sont rapidement apparues. Schlimmer et Fisher [99] proposent ID4 et Utgoff [109] propose ID5R qui sont basés sur ID3 (méthode sur laquelle est basée C4.5) mais dont la construction est incrémentale. En 1997, Utgoff [110] propose un nouvel arbre incrémental, ITI (Incremental Tree Induction), dont le fonctionnement est fondé sur le maintien de statistiques dans les noeuds finaux permettant de restructurer l'arbre lors de l'ajout de nouvelles observations.

Système d'inférence floue : Les systèmes d'inférence floue sont particulièrement adaptés et utilisés pour l'apprentissage incrémental [3]. En logique floue, l'appartenance d'une observation à une classe n'est pas binaire, mais floue. C'est à dire que les classes ne sont pas limitées aux deux valeurs {0, 1}, mais peuvent varier dans tout l'intervalle [0,1]. Une observation peut donc appartenir, à différents degrés, à plusieurs ensembles flous. Différentes fonctions pour prédire la classe peuvent être utilisées. Les principales sont la fonction triangulaire, la fonction trapézoïdale ou la fonction gaussienne. Un système d'inférence floue est composé de règles d'inférence structurées pouvant être activées par une observation. Pour prédire la classe de cette observation, les sorties des règles sont combinées en fonction de leur activation. L'apprentissage incrémental d'un système d'inférence floue se fait par modification des règles d'inférence. En particulier, l'ajout de nouvelles classes se fait facilement en créant de nouvelles règles. La plupart des systèmes d'inférence floue évolutifs sont basés sur les systèmes d'inférence floue de Takagi-Sugeno d'ordre un.

La classification incrémentale pourrait s'intégrer dans le cadre de ce travail. En effet, il pourrait être intéressant d'utiliser ces méthodes pour que le modèle évolue en fonction des internautes. Cependant, la problématique de ce travail repose sur le changement des descripteurs des variables du modèle, en supposant que la population ne change pas. Une autre approche considérée dans ce chapitre est le transfert de connaissance.

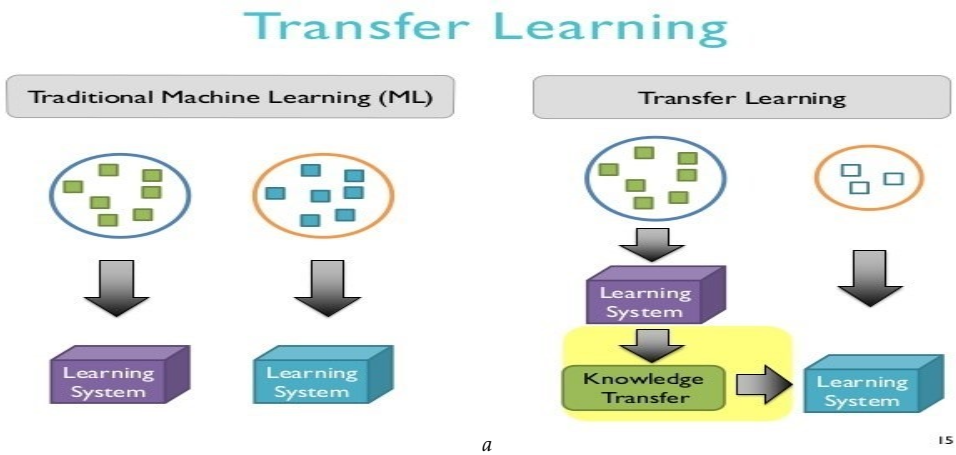
2.2 Transfert de connaissances

Le transfert de connaissances (Transfer Learning) consiste à transférer les connaissances apprises d'un domaine ou d'une tâche "source" vers un domaine ou une tâche "cible". Il se place dans le cadre de l'apprentissage automatique où la plupart des méthodes statistiques et de machine learning reposent sur l'hypothèse que les données des échantillons d'apprentissages et de tests sont tirées d'un même espace de variables et ont la même distribution. De ce fait, lorsque la distribution change, la plupart de ces méthodes doivent être entièrement reconstruites en utilisant les nouvelles données collectées. Le transfert de connaissances affaiblit cette hypothèse en permettant aux domaines, aux tâches et aux distributions utilisées d'être différentes. Ces problèmes d'apprentissages comprennent le *domain adaptation*, *sample biais sélection*, *covariate shift* et *self taught learning*. Ces approches ont pour but commun la réutilisation de connaissances apprises au préalable. Cependant, elles se différencient par des hypothèses qui sont spécifiques à leurs algorithmes d'apprentissages pour gérer le transfert de connaissances.

Notations et définition formelle En reprenant les notations de Pan [82], reprises par Weiss [115], un domaine $\mathcal{D}$ se définit par deux parties : un espace de variables $\mathcal{X}$ et une distribution de probabilité marginale $P(\boldsymbol{x})$. Comme pour la classification, une observation est décrite par d variables caractéristiques notées $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d)$. L'ensemble des variables caractéristiques observées pour chacune des n observations est noté $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$. Les n observations $\mathbf{x}$ sont supposées être des réalisations indépendemment et identiquement distribuées (i.i.d) de $\boldsymbol{X}$. Étant donné un domaine spécifique $\mathcal{D}$, une tâche $\mathcal{T}$ est définie par deux composants : un espace de labels $\mathcal{Z}$ et une fonction prédictive $f(.)$ qui est apprise sur des données d'entraînement. La fonction prédictive définie ici correspond à la règle de décision définie pour la classification, soit $\phi(\boldsymbol{x})$. A l'instar de la classification, la prédiction d'une observation $\boldsymbol{x}$ se fait à l'aide d'un échantillon d'apprentissage composé des couples $(\boldsymbol{x}_i, z_i)$ où $\boldsymbol{x}_i \in \mathcal{X}$ et $z_i \in \mathcal{Z}$. Le transfert de connaissances consiste à appliquer les connaissances d'un domaine ou d'une tâche source à un domaine ou une tâche cible. Un domaine source est alors défini par $\mathcal{D}_S = \{\mathcal{X}_S, P(\boldsymbol{x}_S)\}$ et une tâche source est définie par $\mathcal{T}_S = \{\mathcal{Z}_S, f_S(.)\}$. Symétriquement, un domaine cible est défini par $\mathcal{D}_C = \{\mathcal{X}_C, P(\boldsymbol{x}_C)\}$ et une tâche cible est définie par $\mathcal{T}_C = \{\mathcal{Z}_C, f_C(.)\}$. Les données provenant du domaine source sont décrites par l'échantillon d'apprentissage $D_S = \{(\boldsymbol{x}_{S_1}, z_{S_1}), \ldots, (\boldsymbol{x}_{S_n}, z_{S_n})\}$, où $\boldsymbol{x}_{S_i} \in \mathcal{X}_S$ est une instance de donnée appartenant à l'espace de variable source et $z_{S_i} \in \mathcal{Z}_S$ est le label de la classe correspondante. Symétriquement, les données provenant du domaine cible sont décrites par l'échantillon

d'apprentissage $D_C = \{(\boldsymbol{x}_{C_1}, z_{C_1}), \ldots, (\boldsymbol{x}_{C_n}, z_{C_n})\}$, où $\boldsymbol{x}_{C_i} \in \mathcal{X}_C$ est une instance de donnée et $z_{C_i} \in \mathcal{Z}_C$ est le label de la classe correspondante. Dans la plupart des cas, $0 \leq n_C \ll n_S$, où n_C correspond à la taille de l'échantillon de données cibles et n_S correspond à la taille de l'échantillon de données sources. De la même façon, la tâche source sera notée $\mathcal{T}_S$ et la tâche cible $\mathcal{T}_C$. Dans ce travail, le domaine source $\mathcal{D}_S$ correspond aux données obtenues avant la modification et le domaine cible $\mathcal{D}_C$ correspond aux données obtenues après la modification. La tâche est identique pour les deux domaines et correspond, par exemple, à une classification des internautes selon leurs attentes.

Une définition formelle du transfert de connaissances, reprise de Pan [82] est alors : Étant donné un domaine source $\mathcal{D}_S$ et la tâche source correspondante $\mathcal{T}_S$, un domaine cible $\mathcal{D}_C$ et la tâche cible correspondante $\mathcal{T}_C$, le transfert de connaissances est le processus qui améliore l'apprentissage de la fonction cible $f_C(.)$ dans $\mathcal{D}_C$ en utilisant les connaissances issues de $\mathcal{D}_S$ et $\mathcal{T}_S$, avec $\mathcal{D}_S \neq \mathcal{D}_C$ ou $\mathcal{T}_S \neq \mathcal{T}_C$. La figure 2.1 montre la différence entre le machine learning traditionnel et le transfert de connaissances. On distinguera le transfert de connaissances de l'apprentissage multitâches. En effet, l'apprentissage multitâches consiste en l'amélioration de la performance de toutes les tâches simultanément comme le montre la figure de gauche. Le transfert de connaissances, quant à lui, consiste en l'amélioration de la performance de certaines tâches en utilisant l'information connue d'une tâche auxiliaire ou d'une tâche source, typiquement après que la tâche source ait été apprise.



a

FIGURE 2.1 – Machine Learning vs Transfer learning

a. source : https ://medium.com/data-science-101/

Il existe différentes approches du transfert de connaissances selon ce qui diffère : les

tâches ou les domaines. Est alors définie la notion de transfert de connaissances inductif et transductif [82].

Transfert de connaissances inductif Le transfert de connaissances inductif correspond au transfert de connaissances où les tâches sont différentes. C'est à dire $\mathcal{T}_S \neq \mathcal{T}_C$. Selon la définition d'une tâche, cela implique que soit l'espace des labels est différent $\mathcal{Z}_S \neq \mathcal{Z}_C$, soit les distributions de probabilités conditionnelles sont différentes $P(z_S|\boldsymbol{x}_S) \neq P(z_C|\boldsymbol{x}_C)$. C'est le cas, par exemple, en présence de données déséquilibrées. Dans le cadre de ce travail, les tâches sont les mêmes, ce sont les descripteurs qui changent.

Transfert de connaissances transductif [66] [6] Dans le cas du transfert de connaissances transductif, ce sont les domaines qui sont différents, $\mathcal{D}_S \neq \mathcal{D}_C$. Si l'on reprend la définition précédente, un domaine est défini par deux éléments : $\mathcal{D} = \{\mathcal{X}, P(\boldsymbol{x})\}$. Donc $\mathcal{D}_S \neq \mathcal{D}_C$ implique que, soit les distributions marginales sont différentes $P(\boldsymbol{x}_S) \neq P(\boldsymbol{x}_C)$, soit l'espace des variables est différent $\mathcal{X}_S \neq \mathcal{X}_C$. De nombreux travaux traitent le fait que les distributions marginales soient différentes et se rapportent aux problèmes de *covariates shift*, *sample biais selection* ou encore de *l'adaptation de domaine*. La classification évolutive peut également se placer dans ce cadre. Cependant, ces travaux font l'hypothèse commune que, bien que les distributions marginales soient différentes $\mathcal{D}_S \neq \mathcal{D}_C$, les espaces de variables restent les mêmes $\mathcal{X}_S = \mathcal{X}_C$. Or, dans ce travail, on travaille sur les différences entre les descripteurs de variables, c'est à dire, $\mathcal{X}_S \neq \mathcal{X}_C$. De plus, nous faisons l'hypothèse que les populations entre le domaine source et le domaine cible sont les mêmes, donc nous ne nous plaçons pas dans ce cadre.

Plus récemment, une autre forme de catégorisation du transfert de connaissances est apparue [115]. Celle-ci ne distingue plus le transfert de connaissances en fonction des tâches ou des domaines mais en fonction des différences entre les espaces des variables. Si celui-ci est similaire $\mathcal{X}_S = \mathcal{X}_C$, on parle alors de transfert de connaissances homogènes. Au contraire lorsque les espaces de variables sont différents $\mathcal{X}_S \neq \mathcal{X}_C$ le transfert de connaissances sera dit "hétérogène".

2.2.1 Transfert de connaissances homogène

De nombreuses méthodes reposent sur les hypothèses du transfert de connaissances homogène. En effet, elles regroupent les méthodes du transfert de connaissances inductif et de nombreuses méthodes du transfert de connaissances transductif, notamment les méthodes de covariate shift et d'adaptation de domaine. Selon l'élément qui est transféré,

on peut distinguer 4 types de transferts de connaissances : {le transfert d'instances, le transfert de paramètres, le transfert de variables, le transfert de connaissances des relations entre les domaines}. Ces 4 types de transferts sont maintenant présentés plus en détail.

Transfert d'instances Ce type de transfert trouve ses racines dans les méthodes d'échantillonnages statistiques, où le but est de tirer aléatoirement des instances d'une distribution particulière. Lorsque l'obtention directe des échantillons de la distribution souhaitée est complexe, on approche cette distribution en tirant des échantillons aléatoires provenant d'autres distributions et en les adaptant pour approcher la distribution initiale. Ces algorithmes d'approximations peuvent être utilisés pour corriger les différences entre les distributions des domaines sources et cibles pour le transfert de connaissances. Plus précisément, l'idée intuitive de cette approche est la suivante : même si les tâches sources et cibles sont différentes, il existe une partie des données pouvant être réutilisées, même avec un nombre limité de labels dans la tâche cible. L'idée consiste alors à affecter des poids à certaines instances ou variables du domaine source pour pouvoir l'utiliser dans le domaine cible. C'est cette approche qui est utilisée, par exemple, pour le *covariate shift*.

Covariate Shift initié par Joaquin Quinonero-Candela [89] : En classification classique on suppose que les distributions marginales des espaces source et cible sont similaires $P(\boldsymbol{x}_S) = P(\boldsymbol{x}_C)$. Dans de nombreuses applications cette hypothèse n'est pourtant pas vérifiée (jeu avec des données déséquilibrées, par exemple). Le covariate shift suppose que les distributions marginales sont différentes $P(\boldsymbol{x}_S) \neq P(\boldsymbol{x}_C)$ bien que les distributions conditionnelles restent les mêmes $P(z_S|\boldsymbol{x}_S) = P(z_C|\boldsymbol{x}_C)$. On retrouve cette situation lorsque les données varient au cours du temps ou dans le cas où l'on effectue une sélection d'exemples non i.i.d. comme, par exemple, une sélection de données d'apprentissage artificiellement équilibrée ou encore dans le cadre de l'apprentissage actif.

L'idée générale du transfert de connaissances est l'apprentissage d'un modèle pour la tâche cible qui minimise le risque empirique. Lorsque les distributions sont les mêmes pour les tâches cibles et sources le problème revient à apprendre un modèle en résolvant un problème d'optimisation (minimisation) afin de l'utiliser dans le domaine cible. Lorsque les distributions sont différentes, le problème d'optimisation doit être modifié pour apprendre un modèle consistant à utiliser dans le domaine cible. Cela passe par l'ajout de différents poids d'importance à chaque instance.

— Une solution basique s'apparente à la résolution d'un problème de biais de sélection [54]. Cependant, il peut être compliqué à réaliser dans un espace grande dimension.
— Une autre approche consiste à estimer une fonction de poids $\beta(\boldsymbol{x})$ pour approcher le ratio $\frac{P(\boldsymbol{x}_C)}{P(\boldsymbol{x}_S)}$, qui correspond aux poids pour chaque instance. Ce qui revient à minimiser une fonction objective basée sur la différence des distributions. Les méthodes les plus usuelles sont KLIEP [104],basée sur la distance de Kullback-Leibler ou les méthodes à noyaux telles que (KMM) [57].

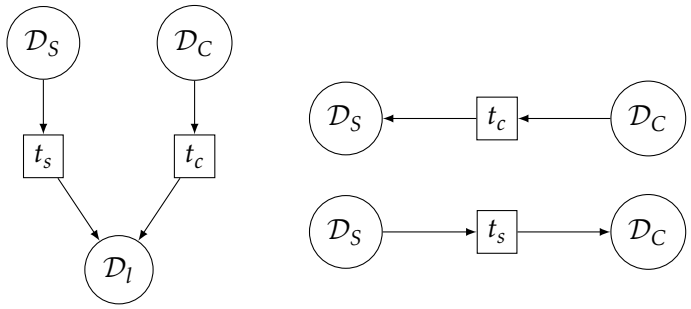
L'avantage de ces méthodes est qu'elles évitent d'estimer les distributions de $\boldsymbol{x}$, qui s'avère être un problème compliqué en grande dimension. Lorsque les distributions conditionnelles sont différentes entre les domaines $P(z_S|\boldsymbol{x}_S) \neq P(z_C|\boldsymbol{x}_C)$, une approche pouvant être employée est l'adaptation jointe des distributions qui consiste à reprendre l'idée de covariate shift en apprenant directement les poids d'importance de la distribution jointe ($P(z,\boldsymbol{x})$) avec un algorithme de style boosting. Cette méthode est reprise par Dai [25] avec l'algorithme TrAdaboost.

Transfert de représentation de variables Ce type de transfert consiste à trouver une représentation des variables de qualité afin de réduire la différence entre les domaines source et cible et dans le même temps réduire l'erreur du modèle de classification ou de régression. Les stratégies pour trouver cette représentation sont différentes selon le type de données de la tâche source. Si les labels du domaine source sont disponibles, une méthode d'apprentissage supervisée est alors utilisée pour construire la représentation. Une fonction de coût à minimiser est alors définie pour évaluer la modélisation des espaces. L'objectif est le même que pour la minimisation des différences entre les distributions marginales, sauf qu'ici on utilise une transformation sur les variables. Selon la méthode utilisée ce type de transfert peut être divisé en deux sous-catégories.

Transfert de connaissances symétrique Le transfert de connaissance symétrique consiste en la découverte d'une structure sous-jacente significative entre les domaines afin de trouver un espace de variables latent. Le but étant que cet espace de variable latent ait une bonne qualité prédictive et qu'il réduise les différences des distributions marginales entre les domaines. A noter, dans cette approche, les domaines sources ($\mathcal{D}_S$) et cibles ($\mathcal{D}_C$) sont modifiés séparément pour ensuite être réunis dans un nouvel espace ($\mathcal{D}_l$) [16]. Cette approche est décrite par la figure 2.2a

Transfert de connaissance asymétrique Le transfert de connaissance asymétrique,

quant à lui, correspond en la transformation des variables sources à l'aide d'une pondération afin de correspondre au mieux au domaine cible, c'est à dire, que l'on modifie les variables du domaine source pour aligner les deux domaines. Cette approche est décrite par la figure 2.2b



a **b**

FIGURE 2.2 – Exemple de transformation symétrique (t_s et t_c) du domaine source $\mathcal{D}_S$ et cible $\mathcal{D}_C$ dans un espace de variables latent commun $\mathcal{D}_l$ (**a**) et d'une transformation asymétrique t_s du domaine source $\mathcal{D}_S$ vers le domaine cible $\mathcal{D}_C$ (**b**)

Par exemple, Long [70] reprend l'idée d'adapter conjointement les distributions mais en passant par un changement de représentation des variables, notamment par une réduction de dimension des variables grâce à une ACP. En plus d'être utile pour la réduction de dimension, l'ACP est également utilisée pour l'optimisation. De plus, les auteurs mesurent la différence entre les distributions marginales avec la mesure MMD (maximum mean discrepancy) qui est intégrée à l'algorithme d'optimisation ACP. Cette méthode est une méthode asymétrique, cependant elle se place dans le cadre homogène, donc, considère que les espaces de variables sont similaires $\mathcal{X}_S = \mathcal{X}_C$.

Plusieurs autres méthodes ont été proposées, cependant, elles ne seront pas détaillées dans cette partie car les espaces des variables de notre travail ne sont pas égaux $\mathcal{X}_S \neq \mathcal{X}_C$.

Transfert de paramètres L'objectif de ce transfert est de découvrir des paramètres ou a priori partagés entre les domaines source et cible, pouvant être bénéfiques pour le transfert de connaissances. Différentes méthodes ont été proposées telles que des méthodes paramétriques. Par exemple, dans [11] les auteurs appliquent une fonction de lien paramétrique entre deux domaines, où les instances ont différentes origines, mais où l'espace des variables reste le même. De même, dans [73] les auteurs proposent également une fonction de lien mais qui ici est une fonction de lien linéaire. L'approche est similaire, cependant les paramètres des différents clusters sont estimés simultanément par un lien

stochastique linéaire. Dans les articles de Bouveyron [19], [18], les auteurs proposent de réutiliser un modèle de régression sur des populations différentes mais ayant les mêmes prédicteurs . Par exemple : prédire le prix des maisons d'une ville de Californie en adaptant un modèle de régression appris sur des données venant d'une autre ville située en Alabama [18]. De même, dans [58], réalisé sur des données binaires, les auteurs recherchent les liens entre deux populations où les paramètres des variables sont différents mais les variables et modalités des variables sont les mêmes Plus récemment, la méthode utilisée pour les données binaires et continues a été adaptée aux modèles mixtes multinomiaux sur des données catégorielles pour pouvoir réaliser un clustering incrémental [52]. Des méthodes utilisant les SVM ont également été proposées telle que la méthode MMKT de Tommasi [107] où l'information transférée est l'hyperplan du SVM.

Relational-knowledge-transfer : Cette approche consiste à transférer des connaissances basées sur les relations (points communs) entre les domaines source et cible. C'est-à- dire créer une cartographie basée sur les relations qu'ont les instances entre les deux domaines. Par exemple, classer les mots d'un document texte en 3 classes en se basant sur la structure de la phrase et sa structure grammaticale. C'est ce qui est fait par Li [69], par exemple.

Ces approches sont relatives au transfert de connaissances homogène, cependant, l'hypothèse que les espaces de variables soient identiques $\mathcal{X}_S = \mathcal{X}_C$, limite leurs applications pour de nombreux problèmes réels. Pour pallier à cette limite, des méthodes de transfert de connaissances hétérogène peuvent être utilisées. En effet, le transfert de connaissances hétérogène repose sur l'hypothèse que les espaces de variables du domaine source et cible sont différents $\mathcal{X}_S \neq \mathcal{X}_C$.

2.2.2 Transfert de connaissances hétérogène (HTL)

Le transfert de connaissances hétérogène est apparu assez récemment (il y a 6 -7 ans) et a des applications dans de nombreux domaines, notamment liés au web. Le but du transfert de connaissances hétérogène est de créer un lien (pont) entre les espaces de variables. Alors qu'avec le transfert de connaissances homogène, quatre types de transfert peuvent être utilisés, le transfert de connaissances hétérogène n'en utilise qu'une. Comme le but du transfert de connaissances hétérogène est de créer un lien entre les espaces de variables, celui-ci utilise donc le transfert de connaissances basé sur la représentation des variables, présenté pour le transfert de connaissances homogène.

De la même manière que pour le transfert de connaissances homogène, on distinguera l'approche symétrique de l'approche asymétrique. Dans notre problème, l'objectif est d'utiliser toute l'information disponible avec des paramètres interprétables en ce qui concerne le changement des descripteurs de variables. Le but étant d'utiliser l'ensemble des données du domaine source et du domaine cible pour réaliser des analyses statistiques. Nous nous situons alors dans le cadre de l'approche asymétrique. Cependant, une méthode se plaçant dans le cadre symétrique a retenu notre attention. La méthode, proposée par Dai [24], Translated Learning via Risk Minimization.

Translated Learning via Risk Minimization(TLRisk) Dans cette méthode, les auteurs proposent d'effectuer une forme de mapping asymétrique des variables venant d'un espace source vers des variables venant d'un espace cible, afin d'effectuer l'apprentissage dans un seul espace de variables latent. La modélisation des liens entre les variables est effectuée par un traducteur créé à l'aide du "language model". L'idée est de combiner la traduction de variable et l'apprentissage des plus proches voisins dans un modèle unifié en utilisant la méthode du "language model". La modélisation proposée dans cette méthode est une chaine de Markov $k \rightarrow \boldsymbol{x}_s \rightarrow \mathbf{x}_s$ pour l'apprentissage dans l'espace source, où $\mathbf{x}_s = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ et k représente la k^{eme} classe. Cette chaîne peut être connectée à une autre chaîne de Markov $k \rightarrow \boldsymbol{x}_c \rightarrow \mathbf{x}_c$ dans l'espace cible. Le transfert de connaissances de l'espace source vers l'espace cible peut alors être modélisé par $z \rightarrow \boldsymbol{x}_s \rightarrow \boldsymbol{x}_c \rightarrow \mathbf{x}_c$, où le lien $\boldsymbol{x}_s \rightarrow \boldsymbol{x}_c$ est utilisé comme niveau de variable traducteur. La traduction est réalisée par l'apprentissage d'un modèle probabiliste utilisant les données co-occurrentes comme un pont entre les espaces de variables source et cible. Pour connecter deux espaces de variables différents, les auteurs calculent la probabilité des variables cibles sachant les variables sources : $P(\boldsymbol{x}_c|\boldsymbol{x}_s)$. Pour estimer cette probabilité, les données co-occurrentes sont alors nécessaires. L'estimation de la probabilité conditionnelle du traducteur est alors réalisée par :

$$P(\boldsymbol{x}_c|\boldsymbol{x}_s) = \frac{P(\boldsymbol{x}_c, \boldsymbol{x}_s)}{\int P(\boldsymbol{x}'_c, \boldsymbol{x}_s) d\boldsymbol{x}'_c}. \tag{2.6}$$

La probabilité jointe est donnée par les données ou variables co-occurrentes. Par ailleurs, les auteurs se placent dans un cadre de minimisation de risque [42] pour évaluer la précision de l'apprentissage utilisant les données d'entraînement des deux domaines et la fonction de traduction. Comme la fonction de risque à minimiser utilisée peut être difficile à estimer car c'est une intégration sur

l'ensemble des espaces de modèles possibles, les auteurs proposent d'estimer cette fonction en l'approchant par une mesure de distance entre les modèles. Le temps de calcul de ces opérations étant coûteux, l'algorithme est implémenté en programmation dynamique.

Cette méthode se place dans un cadre probabiliste et les données co-occurrentes sont nécessaires pour avoir des probabilités jointes entre les espaces et estimer les probabilités conditionnelles. Dans le cadre de notre travail, cette méthode est intéressante car elle se place dans un cadre probabiliste pour réaliser le transfert de connaissances. Cependant, dans cette méthode, il n'y a pas de données manquantes et les probabilités jointes sont connues, grâce à l'utilisation de données co-occurrentes. D'autre part, notre travail se place plutôt dans le cadre du transfert de connaissance asymétrique.

Méthodes asymétriques Les méthodes proposées de l'approche asymétrique se distinguent en fonction des jeux de données auxquelles elles s'adressent. En effet, certaines ont besoin des labels dans les deux domaines (même si en faible quantité pour le domaine cible), d'autres des labels d'un seul des deux domaines ou même d'aucun label des deux domaines. De plus, certaines méthodes requièrent de multiples sources de données où concernent des problèmes multi-tâches. Le cadre de notre travail implique d'avoir les labels du domaine source et a minima quelques labels pour le domaine cible. De plus, nous n'avons qu'une seule source de données et qu'une seule tâche de classification. Nous allons maintenant détailler quelques méthodes qui se placent dans le cadre de notre problématique et se rapprochent de notre contexte. Ce sont les méthodes {SHFR, ARC-t, IFSR et CTLearn}

Asymmetric regularized cross-domain Transformation (ARC-t) Dans cette méthode proposée par Kullis [67], l'auteur propose un algorithme de transformation asymétrique pour résoudre l'espace de variables hétérogènes entre les domaines. Cette méthode requiert d'avoir de nombreux labels pour le domaine source et une quantité de labels limitée pour le domaine cible. ARC-t repose sur le principe utilisé par la méthode (ITML : Information Theoretic Metric Learning) proposée par Saenko [97] dans le cadre du transfert de connaissances homogènes. ITML peut être vue comme l'apprentissage d'une transformation linéaire $W \in \mathbb{R}^{d_s * d_c}$ entre le domaine source D_s et le domaine cible D_c. Cet apprentissage est optimisé pour satisfaire des contraintes entre les points transformés qui sont exprimés comme une fonction : $\mathbf{x}_s^T W \mathbf{x}_c$ avec $\mathbf{x}_s \in D_s$ et $\mathbf{x}_c \in D_c$. Les auteurs appliquent leur méthode dans un espace à noyau afin d'avoir une transformation non-linéaire. La méthode

ITML s'applique uniquement dans le cas homogène car la régularisation utilisée sur W est *LogDet*, ce qui implique que les dimensions des espaces source et cible soient équivalentes. Les auteurs de la méthode ARC-t étendent la méthode ITML par une transformation asymétrique. Dans un premier temps, ils définissent, dans un cas non contraint, une fonction objectif : $\min_W r(W) + \lambda \sum_i c_i(X_s^T W X_c)$, où X_s est la matrice des points de D_s, X_c est la matrice des points de D_c, r est la matrice de régularisation et c_i une fonction de coûts sur les contraintes. Cette fonction objectif a pour but de trouver la matrice de transformation. Le but est alors de minimiser la matrice r et un jeu de contraintes reposant sur la similarité ou dissimilarité des paires $(\mathbf{x}_s, \mathbf{x}_c)$. Pour surpasser les limitations induites par la fonction objectif présentée ci-dessus, les auteurs se placent dans un espace à noyau, ce qui leur permet de ne pas être limité à des transformations linéaires de W et où la complexité est indépendante des dimensions de $\mathcal{X}_s$ et $\mathcal{X}_c$. Contrairement à la méthode ITML, les contraintes sont définies par des fonctions telles que : $c_i(X_s^T W X_c) = (max(0, l - \boldsymbol{x}_s^T W \boldsymbol{x}_c))^2$ si $\boldsymbol{x}_s$ vient de la même catégorie que $\boldsymbol{x}_c$ et $c_i(X_s^T W X_c) = (max(0, \boldsymbol{x}_s^T W \boldsymbol{x}_c - u))^2$ si $\boldsymbol{x}_s$ ne vient pas de la même catégorie que $\boldsymbol{x}_c$. Où l et u sont les bornes supérieures et inférieures choisies pour mesurer la similarité $\boldsymbol{x}_s^T W \boldsymbol{x}_c$.

Cette méthode se place dans le cadre des espaces à noyau alors qu'à la vue de nos données, nous nous plaçons dans un cadre probabiliste. De plus, le calcul des contraintes de similarité/dissimilarité des catégories entre les domaines source et cible suppose que les couples $(\mathbf{x}_S, \mathbf{x}_C)$ et leurs appartenances aux catégories soient connus. Or, notre objectif est d'estimer ces liens.

Sparse heterogeous feature representation (SHFR) Proposée par Zou [117], cette méthode requiert d'avoir une grande quantité de données labélisées du domaine source et une petite quantité de labels du domaine cible. Comme pour la méthode précédente, elle utilise une matrice de transformation $W \in \mathbb{R}^{d_s * d_c}$ pour lier les données du domaine cible aux données du domaine source. Cependant, l'apprentissage de la transformation est réalisé différemment. Notamment, cette méthode repose sur deux hypothèses :

— La matrice de transformation W utilisée entre les deux domaines est creuse
— La transformation est invariante aux classes, ce qui signifie que toutes les classes partagent la même configuration.

L'appariement entre les variables du domaine source et du domaine cible est réalisé à l'aide d'une fonction de transformation asymétrique. L'idée est de reconstruire une matrice creuse de transformation des variables qui est apprise à l'aide

d'une méthode d'apprentissage multi-tâches (Ando [5]). Un classifieur binaire est implémenté pour chaque classe dans le domaine source et cible séparément. Il est supposé que le classifieur est linéaire, soit pour le domaine source, soit pour le domaine cible, de manière à avoir $f^t(\boldsymbol{x}) = \mathbf{w}^{tT}\boldsymbol{x}$, où $\mathbf{w}^t$ est le vecteur de poids pour le classifieur t et t le classifieur binaire correspondant à la classe k. La matrice de transformation est alors apprise, soit en maximisant les dépendances entre les vecteurs de poids transformés, associés aux classifieurs sources, et les vecteurs de poids associés aux classifieurs cibles : $\max_W \mathbf{w}_c^{tT} W \mathbf{w}_s^t$, où W est la matrice de transformation. Soit en minimisant la distance entre les deux vecteurs de poids : $\min_W \|\mathbf{w}_c^t - W\mathbf{w}_s^t\|$. En utilisant la seconde méthode, la relation entre le vecteur de poids source et le vecteur de poids cible peut être modélisée par $\mathbf{w}_c^t - W\mathbf{w}_s^t = \mathbf{w}_\Delta^t$, avec $\mathbf{w}_\Delta^t$ défini comme un vecteur de poids et où sa norme l_2 peut être utilisée pour mesurer la différence entre le vecteur de poids cible et le vecteur de poids transformé source. Afin d'avoir un vecteur de poids robuste pour réaliser les prédictions dans le domaine cible, la matrice de transformation W est apprise en minimisant $\|\mathbf{w}_\Delta^t\|_2$. Finalement, comme la matrice W doit être creuse et de classe invariante, une optimisation jointe de W sur toutes les tâches binaires peut être réalisée.

Contrairement à nous, cette méthode ne se place pas dans un cadre probabiliste.

Informed Feature Space Remapping (IFSR) L'idée de cette méthode proposée par Feuz et Cook [36] est de construire des méta-variables pour le transfert de connaissances plutôt que d'utiliser les données co-occurrentes. Plus globalement, les auteurs proposent une transformation asymétrique pour relier les données d'un domaine cible aux variables d'un domaine source. Pour définir leur problème les auteurs utilisent l'hypothèse $H_s : \mathcal{X}_s \rightarrow Z_s$. Le problème est alors de trouver une modélisation $\theta(\mathcal{X}_s, \mathcal{X}_c)$ qui minimise l'erreur :$error_\theta(H_s)$ où $error_\theta(H_s)$ représente l'erreur empirique du domaine cible en utilisant H_s sur les données appariées du domaine cible. Dans un premier temps, les méta-variables sont trouvées en calculant la valeur co-occurrente de la variable label pour chaque variable et label des espaces source et cible. C'est à dire en calculant la valeur de l'espérance des variables basée sur les labels connus des données d'entraînement labélisées. Donc, si $Z = Z_s \cup Z_c$ chaque donnée co-occurrente pour chaque variable et chaque label est calculée comme :

$$E(\boldsymbol{x}|k) = \frac{1}{n_k}\sum_{i=1}^{n} \boldsymbol{x}_i \tag{2.7}$$

où k est le label et n_k le nombre de données ayant le label k. Ce calcul est supposé dans un espace à valeurs réelles mais peut être étendu aux valeurs catégorielles en calculant le nombre d'occurrences de chaque catégorie pour estimer la probabilité.

Une fois les méta-variables calculées, les auteurs construisent un score basé sur un classifieur de Bayes naïf pour chaque paire de variables. Une matrice de similarité entre les paires de variables est alors construite en utilisant le score donné par les méta-variables. Le lien est alors créé en sélectionnant la variable source x_s ayant le score de similarité maximal avec la variable x_c, donné par la matrice de similarité. Les liens générés sont *many to one*, c'est à dire qu'une seule variable source peut être liée à une variable cible, mais plusieurs variables cibles peuvent être liées à une même variable source. Enfin, les auteurs appliquent le mapping sur les données cibles pour apprendre le classifieur en utilisant l'hypothèse apprise sur les données sources.

L'idée de cette méthode se rapproche de l'objectif que nous avons. C'est-à-dire retrouver le lien entre les paires de variables sources et cibles. Cependant, dans cette méthode, il n'y a pas de données supposées manquantes à estimer. De plus, la méthode ne se place pas dans le cadre probabiliste, et n'estime pas de probabilités jointes comme nous souhaitons le faire.

Co-transfert learning via joint transition probability graph based method (CT-Learn) [81] Dans cette méthode, une stratégie de co-transfert learning est utilisée pour construire les liens entre les espaces de variables à travers des données co-occurrentes. On notera que l'objectif principal de cette méthode est de réaliser un transfert de connaissances sur de multiples espaces et non seulement deux espaces. L'idée des auteurs est alors de modéliser leur problème comme un graphe de probabilités de transition jointes de toutes les instances (sources et cibles). Cela leur permet d'utiliser l'idée proposée par [53] et la marche aléatoire avec recommencement [108]. Les probabilités de transition sont construites en utilisant *l'intra-relation* basée sur une métrique d'affinité entre les instances et *l'inter-relation* basée sur les données co-occurrentes des instances venant des différents espaces. Le calcul de la métrique d'affinité entre les instances est basé sur la différence entre les vecteurs de variables sources et cibles, représentée par la norme$_2$. Du fait de la structure intrinsèque des données, les auteurs utilisent une fonction gaussienne à noyau afin d'avoir une version non linéaire de l'affinité.

La métrique d'affinité peut alors être écrite par :

$$a_{k,l}^{(i,i)} = exp\left[\frac{-\|\boldsymbol{x}_k^{(i)} - \boldsymbol{x}_l^{(i)}\|_2}{2\sigma^2}\right] \tag{2.8}$$

où σ est un nombre positif qui contrôle le lien dans la structure et i est l'espace de variables. La métrique d'affinité est construite pour un même espace de variables. Une matrice carrée $\mathbf{A}^{(i,i)}$ est alors construite, représentant l'intra-relation. Une seconde matrice $\mathbf{P}^{(i,i)}$ est également construite par normalisation de $\mathbf{A}^{(i,i)}$ afin d'être utilisée comme matrice de transition Markovienne.

Une seconde matrice $\mathbf{A}^{(i,j)}$ est ensuite construite avec les données co-occurrentes de deux espaces différents, i et j indiquant les espaces. De la même manière que pour la matrice d'affinité, une matrice $\mathbf{P}^{(i,j)}$ est construite en normalisant la matrice $\mathbf{A}^{(i,j)}$. Comme il est possible que certaines colonnes de $\mathbf{P}^{(i,j)}$ soient égales à 0 si il n'existe pas de données co-occurrentes, les entrées de la colonne sont fixées à $\frac{1}{n_i}$. Le but étant que lors de la marche aléatoire, chaque état puisse être visité de façon équiprobable. Le graphe de probabilités jointes est alors construit en utilisant les deux matrices $\mathbf{P}^{(i,i)}$et $\mathbf{P}^{(i,j)}$ et en utilisant un modèle de chaîne de Markov couplée. La matrice du graphe de transition joint est alors de la forme :

$$\mathbf{P} = \begin{pmatrix} \lambda_{1,1}\mathbf{P}^{(1,1)} & \lambda_{1,2}\mathbf{P}^{(1,2)} & \dots & \lambda_{1,N}\mathbf{P}^{(1,N)} \\ \lambda_{2,1}\mathbf{P}^{(2,1)} & \lambda_{2,2}\mathbf{P}^{(2,2)} & \dots & \lambda_{2,N}\mathbf{P}^{(2,N)} \\ \vdots & \vdots & \vdots & \\ \lambda_{N,1}\mathbf{P}^{(N,1)} & \lambda_{N,2}\mathbf{P}^{(N,2)} & \dots & \lambda_{N,N}\mathbf{P}^{(N,N)} \end{pmatrix}$$

où le paramètre pondérant $\lambda_{i,j}$ contrôle la quantité de connaissance transférée du j^{eme} espace au i^{eme} espace pendant le processus d'apprentissage. Par exemple, pour un transfert de connaissance binaire, la connaissance est transférée uniquement de la source vers la cible mais pas l'inverse. En utilisant cette matrice, une observation visite itérativement ses noeuds voisins. L'observation a une probabilité stationnaire de finalement rester dans un noeud différent. Les auteurs utilisent alors cette probabilité pour calculer un score de rang du label pour indiquer l'importance du label pour une instance test. Les probabilités stationnaires sont représentées dans une matrice $\mathbf{U}$.

On notera que l'algorithme est implémenté avec une programmation dynamique

où il est uniquement nécessaire de calculer la matrice **U** afin d'avoir le score de rang. La chaîne de Markov couplée est alors formulée pour le processus d'apprentissage contrairement à la méthode TLRisk [24] qui utilise la chaîne de Markov pour estimer les paramètres.

Cette méthode est très proche de la nôtre, on peut cependant noter quelques différences. Le problème de co-transfert de connaissances est différent du problème de transfert de connaissances où les espaces sources et cibles sont habituellement fixés. Or, nous sommes dans un problème de transfert de connaissances. De plus, dans cette méthode, il est nécessaire de connaître la connectivité entre les différents espaces, ce qui suppose qu'il n'y a pas de données manquantes. En outre, la matrice de transition **P** n'est pas stochastique et n'a pas d'entrée à 0.

2.3 Conclusion

Dans cette partie, nous avons vu l'ensemble des approches de classification disponibles. Étant donné notre problématique, nous nous plaçons dans le cadre du transfert de connaissances hétérogène. Cependant, aucune des méthodes existantes ne semble traiter l'ensemble de notre problématique, même si certaines s'en rapprochent fortement. Le chapitre suivant présente notre modélisation du problème, intégrant les données manquantes. Il est à noter que notre approche sera totalement indépendante de la méthode de classification, de la même façon que pour la méthode SHFR. Le chapitre suivant présente notre approche.

Chapitre 3

Transfert de connaissances entre espaces qualitatifs de dimensions différentes

> Notre connaissance de ce qui sera est en raison de notre connaissance de ce qui est et de ce qui fut. La science est prophétique. Plus une science est exacte, plus on en peut tirer d'exactes prophéties.
>
> Citation de Anatole France ; Sur la pierre blanche (1905)

Dans ce chapitre nous présentons la modélisation proposée pour répondre au problème étudié. L'objectif principal de ce travail est la modélisation des liens pouvant se former entre les descripteurs de deux variables catégorielles. Un second objectif inhérent à la modélisation proposée est d'avoir un modèle interprétable afin de comprendre l'impact d'une modification et de pouvoir utiliser le modèle pour répondre à différents objectifs (classification, typologie, ...). Contrairement aux méthodes vues dans le chapitre précédent, notre approche se focalise sur une seule variable catégorielle à la fois et non sur l'ensemble des variables disponibles. D'autre part, les méthodes présentées dans le chapitre précédent supposent que les échantillons n'ont pas de données manquantes. Or, il a été présenté dans la section 1.2 que les échantillons collectés pour notre problème comportaient de nombreuses données manquantes.

3.1 Rappel de la problématique

3.1.1 Objectif et problématique

L'objectif initial de ce travail est de pouvoir réaliser différentes analyses statistiques, telle que de la classification. Cependant, comme indiqué dans le chapitre 2, les données disponibles ne permettent pas l'application des méthodes classiques d'analyse de données. En effet, la transformation d'une variable qualitative implique de travailler avec des espaces de dimension différente avant et après la modification et donc la refonte des différentes analyses de données. Reprenant le chapitre 1, nous supposons que les deux versions de la question, correspondant à la variable étudiée, sont remplies par le même nombre d'individus n dont l'ensemble des réalisations correspondent aux variables $\boldsymbol{x}$ et $\boldsymbol{y}$. Les n individus sont ensuite séparés en deux groupes de tailles respectives n^- et n^+, avec $n = n^- + n^+$ représentant les périodes avant et après modification de la question.

Période avant la redéfinition du site web n^- internautes ont renseigné la variable (ou question) $\boldsymbol{x}$, produisant des réalisations *observées* $\mathbf{x}^- = (\boldsymbol{x}_1^-, \ldots, \boldsymbol{x}_{n^-}^-)$. Comme on vient de le préciser, la variable $\boldsymbol{y}$ n'a jamais été renseignée par contre, produisant des réalisations *non observées* $\mathbf{y}^- = (\boldsymbol{y}_1^-, \ldots, \boldsymbol{y}_{n^-}^-)$.

Période après la redéfinition du site web De façon symétrique, n^+ internautes ont renseigné la variable $\boldsymbol{y}$, produisant des réalisations *observées* $\mathbf{y}^+ = (\boldsymbol{y}_1^+, \ldots, \boldsymbol{y}_{n^+}^+)$. Comme attendu, la variable (ou question) $\boldsymbol{x}$ n'a jamais été renseignée par contre, produisant des réalisations *non observées* $\mathbf{x}^+ = (\boldsymbol{x}_1^+, \ldots, \boldsymbol{x}_{n^+}^+)$.

Une partition $\mathbf{x} = (\mathbf{x}^-, \mathbf{x}^+)$ et $\mathbf{y} = (\mathbf{y}^-, \mathbf{y}^+)$ est alors réalisée, et est simplement notée $\mathbf{x}^-, \mathbf{x}^+, \mathbf{y}^-$ et $\mathbf{y}^+$. Ce travail repose sur l'hypothèse que les distributions marginales de $\boldsymbol{x}$ et $\boldsymbol{y}$ ne changent pas au cours du temps. C'est à dire que la population, au sens statistique, reste la même avant et après la modification. Sous cette hypothèse, et le nombre d'observations n^- étant conséquent, le nombre de réalisations observées $\mathbf{x}^-$ permet d'avoir une bonne connaissance de la distribution de la variable $\boldsymbol{x}$ et ne pose pas de problème pour la réalisation des différentes analyses statistiques. Cependant, la question ayant été modifiée, l'objectif est désormais d'utiliser les réalisations observées $\mathbf{y}^+$ dans les différentes analyses, dont l'espace est de dimension différente de l'espace des réalisations $\mathbf{x}$. Cela implique de refaire les différentes analyses utilisant la question, tels que les modèles de classifications. La refonte des différentes analyses utilisant la question modifiée nécessite d'attendre un temps non négligeable pour récolter une grande quantité de données, ou de travailler rapidement avec des échantillons de petites tailles. Le contexte industriel de ce travail nécessite de travailler rapidement avec des

échantillons de petites tailles. Les méthodes classiques d'analyses de données, tels que la classification, ne sont alors pas utilisables. En effet, les hypothèses fortes sur lesquelles elles reposent, présentées dans le chapitre 2, ne sont pas respectées. La faible quantité de données pour la variable $\boldsymbol{y}$ implique une très faible connaissance de cette variable. Afin d'avoir une meilleure connaissance de la variable $\boldsymbol{y}$, rapidement, nous optons donc pour une approche de transfert de connaissances, également présentée dans le chapitre 2. L'objectif est alors d'utiliser la connaissance de la variable $\boldsymbol{x}$, qui correspond à notre espace source, pour la transférer à la variable $\boldsymbol{y}$, correspondant à l'espace cible. Ce transfert de connaissances permettrait l'utilisation de la variable $\boldsymbol{y}$ pour la réalisation des différentes analyses de données nécessitant de grands échantillons. D'autre part, dans le chapitre 1, un objectif d'interprétation des impacts de la modification a également été formulé. Le transfert des connaissances de la variable $\boldsymbol{x}$ vers la variable $\boldsymbol{y}$ peut également permettre de répondre à cet objectif. L'hypothèse que les distributions marginales de $P(\boldsymbol{x})$ et $P(\boldsymbol{y})$ ne changent pas au cours du temps implique que nous nous plaçons dans le cadre du transfert de connaissance transductif et plus particulièrement dans la cadre du transfert de connaissances hétérogène, détaillé dans le chapitre 2. En effet, les espaces source et cible étant de dimension différentes, l'espace source correspondant à l'espace de la variable $\boldsymbol{x}$ est différent de l'espace cible, correspondant à l'espace de la variable $\boldsymbol{y}$.

3.1.2 Formalisation de la problématique et notations

Cette partie reprend la formalisation définie dans la section 1.1. Un questionnaire dit « Questionnaire dynamique » correspond à un couple de questionnaires (respectivement $Q_{\mathbf{x}}$ et $Q_{\mathbf{y}}$), où tous deux sont remplis par le même nombre n d'individus (l'ensemble des résultats sont respectivement les variables $\mathbf{x}$ et $\mathbf{y}$). Ces variables ont la même signification pour les deux questionnaires mais pas nécessairement le même nombre de modalités (p et q respectivement). Il est possible que $p = q$, cela signifie que les modalités ont une signification différente que l'on va chercher à apparier. Dans ce chapitre une notation binaire est adoptée. Classiquement, cette notation s'écrit de la façon suivante : $\mathbf{x} = (\boldsymbol{x}_i)_{i=1,\dots,n}$ avec $\boldsymbol{x}_i = \boldsymbol{x}_{ij} = (x_{ijh})_{h=1,\dots,p}$, où l'individu $x_{ijh} = 1$ si l'individu a sélectionné la modalité h de la variable j, $x_{ijh} = 0$ sinon. Cependant, le cadre de ce travail implique une focalisation sur le changement de descripteurs d'une seule variable catégorielle à la fois. Cette notation est alors simplifiée, rendant implicite l'indice de la variable j. La notation adoptée dans la suite de ce travail est alors la suivante : Soit $\mathbf{x}$ le vecteur des réalisations de la variable $\boldsymbol{x}$ où $\mathbf{x} = (\boldsymbol{x}_i)_{i=1,\dots,n}$ avec $\boldsymbol{x}_i = (x_{ih})_{h=1,\dots,p}$, où l'individu $x_{ih} = 1$ si l'individu a sélectionné la modalité h de l'élément dans $\boldsymbol{x}$, $x_{ih} = 0$ sinon. Symétriquement,

$\mathbf{y} = (\boldsymbol{y}_i)_{i=1,\dots,n}$ avec $\boldsymbol{y}_i = (y_{ih'})_{h'=1,\dots,q}$, où l'individu $y_{ih'} = 1$ si l'individu a sélectionné la modalité h' de l'élément dans $\boldsymbol{y}$, $y_{ih'} = 0$ sinon.

3.1.3 Exemple chez MeilleureAssurance

Afin d'introduire notre modélisation générale, un cas d'usage venant de la société MeilleureAssurance.com a été utilisé. La question étudiée est la suivante : Comment les internautes réagissent-ils lorsque les descripteurs d'une variable changent ? Pour répondre à cette question, nous nous focalisons sur la variable « Niveau de garantie souhaité » que les internautes doivent renseigner. Cette variable est présentée dans la section 1.1.4 où la figure 1.4 montre les modifications dont elle a fait l'objet à plusieurs reprises. Dans cette partie, nous nous concentrons uniquement sur le changement initial du 09/09/2014. Les autres changements effectués sur cette variable serviront par la suite pour la validation de la méthode proposée. Initialement, cette variable avait quatre choix de réponses possibles (ou descripteurs) étant {Tiers (T), Tiers++ (T++), Intermédiaire (I), Tous Risques (TR)}. Dans cet exemple p est donc égal à 4. Dans un second temps, suite à un redéfinition du site web, cette variable a été décomposée en sept nouveaux descripteurs qui sont {Tiers (T), Tiers+ (T+), Tiers++ (T++), Intermédiaire (I), Tous Risques (TR), Tous Risques+ (TR+), Tous Risques++ (TR++)}. Suivant les notations précédentes, dans cet exemple $q = 7$. La figure 3.1 est une représentation de la modélisation de cet exemple où les arcs orientés représentent les transitions possibles entre les descripteurs.

3.2 Résolution par l'estimation des liens entre les modalités des variables x et y

3.2.1 Les probabilités de transition comme clé du problème

Idéalement, la solution de ce problème serait de transposer les réalisations $\mathbf{x}$ de la variable $\boldsymbol{x}$ en réalisations $\hat{\mathbf{y}}$ de la variable $\boldsymbol{y}$, dans la but d'augmenter la taille de l'échantillon $\mathbf{y}$. La quantité d'observations disponibles pour la variable $\boldsymbol{y}$ étant trop faible pour la réalisation des différentes analyses statistiques, tel que de la classification, l'augmentation du nombre d'observations dans l'échantillon permettrait la réalisation de ces analyses. Cependant, la modification des descripteurs implique une grande part d'incertitude qu'il est nécessaire de quantifier. Énoncé dans le chapitre 1, le cadre probabiliste est le plus adapté et le plus usuel pour quantifier l'aléatoire. Nous nous placerons donc dans ce cadre pratique pour formaliser explicitement des hypothèses. Par exemple,

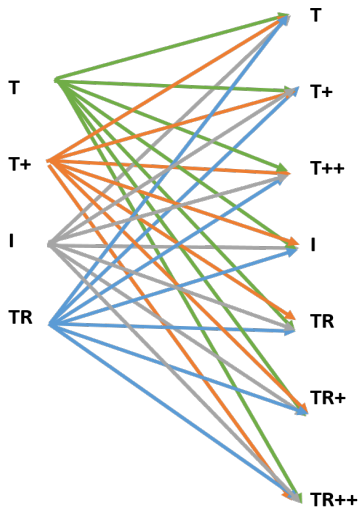


FIGURE 3.1 – Graphe modélisant l'exemple et les transitions possibles entre les variables x et y

nous supposons que les variables x et y sont des variables aléatoires dont les réalisations sont indépendamment et identiquement distribuées. Étant donné qu'une grande quantité d'observations est disponible pour la variable x, il est possible d'avoir une bonne connaissance de sa loi. A l'inverse, la faible quantité d'observations disponibles pour la variable y implique une mauvaise connaissance de sa loi (estimateurs de trop grande variance). L'objectif est ici d'utiliser la loi de x pour générer de nouveaux échantillons $\hat{y}$. Nous nous plaçons alors dans une approche générative où le cadre probabiliste nous permet de générer des observations $\hat{y}$. Ayant déjà une bonne connaissance de la loi de probabilité de x, il est alors nécessaire d'obtenir les probabilités de transition $P(y|x)$ pour la génération de nouveaux échantillons $\hat{y}$. D'autre part, outre la réalisation des diverses analyses statistiques, la connaissance de la loi des probabilités de transition $P(y|x)$ permettrait également de répondre aux différents objectifs d'interprétations concernant les impacts liées aux changement de descripteurs. Enfin, l'utilisation de l'approche générative dans un cadre probabiliste nous permet également de pouvoir mesurer la variabilité de la distribution générée à l'aide d'expérience de Monte Carlo par exemple. Désormais, notre objectif est donc d'obtenir la loi des probabilités conditionnelles $P(y|x)$.

3.2.2 Formalisation du problème

Les données manquantes inhérentes au problème étudié impliquent que l'ensemble des probabilités de transition $P(y|x)$ soit totalement inconnu. En effet, pour chaque

couple d'observation $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, seulement un des membres du couple est connu, tel qu'expliqué dans le chapitre 1.2. Le fait de n'avoir qu'une donnée partielle pour chaque couple d'observations $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ implique alors une absence de connaissance des liens entre les variables $\boldsymbol{x}$ et $\boldsymbol{y}$. La loi jointe permet la modélisation de l'ensemble des liens stochastiques entre les deux variables et permet également de faire apparaître les probabilités de transition $P(\boldsymbol{y}|\boldsymbol{x})$ par la relation suivante :

$$P(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{x}). \tag{3.1}$$

Étant donné que les variables aléatoires $\boldsymbol{x}$ et $\boldsymbol{y}$ sont catégorielles, elles suivent des lois multinomiales d'ordres respectivement p et q. Nous notons alors $\boldsymbol{x}_i \sim M_p(\mathbf{p})$ et $\boldsymbol{y}_i|x_{ih} = 1 \sim M_q(\mathbf{p}_h)$ où $\mathbf{p} = (p_h)_{h=1,\dots,p}$ avec $p_h = P(x_{ih} = 1)$, et où $\mathbf{p}_h = (p_{hh'})_{h'=1,\dots,q}$ avec $p_{hh'} = P(y_{ih'} = 1|x_{ih} = 1)$. Dans la suite, la notation $\mathbf{p}_.$ sera utilisée comme raccourci pour $\{\mathbf{p}_h\}_{h=1,\dots,p}$. Se plaçant dans un cadre probabiliste, tous les couples $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ sont supposés stochastiquement indépendants, conditionnellement à leurs paramètres. La loi jointe $P(\boldsymbol{x}, \boldsymbol{y})$ suit également une loi multinomiale d'ordre pq et peut s'écrire de la façon suivante :

$$P(\mathbf{x}, \mathbf{y}; \mathbf{p}, \mathbf{p}_.) = \prod_{i=1}^{n} P(\boldsymbol{y}_i|\boldsymbol{x}_i; \mathbf{p}_.)P(\boldsymbol{x}_i; \mathbf{p}). \tag{3.2}$$

De nouveau, cette écriture à l'intérêt de faire apparaître les probabilités de transition (ou d'appariement) $\mathbf{p}_.$ entre les descripteurs des variables $\boldsymbol{x}$ et $\boldsymbol{y}$. L'utilisation des lois multinomiales et l'hypothèse d'indépendance des couples d'observations $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ permet l'obtention d'un modèle paramétrique où la log-vraisemblance observée des paramètres $(\mathbf{p}, \mathbf{p}_.)$ (détaillée dans la section 3.4.1) peut être définie et formulée de la façon suivante :

$$\ell(\mathbf{p}_., \mathbf{p}; \mathbf{x}^-, \mathbf{y}^+) = \sum_{h'=1}^{q} \sum_{i=1}^{n^+} y_{ih'}^+ \ln\left(\sum_{h=1}^{p} p_{hh'} p_h\right) + \sum_{h=1}^{p} \sum_{i=1}^{n^-} x_{ih}^- \ln p_h. \tag{3.3}$$

L'ensemble des probabilités de transition $\mathbf{p}_.$ et les probabilités $\mathbf{p}$ étant des paramètres inconnus, la problématique est désormais d'estimer toutes ces probabilités. La figure 3.1 indique un grand nombre de probabilités à estimer, notamment un grand nombre de probabilités de transition. D'autre part, les données disponibles correspondant aux lois marginales $P(\boldsymbol{x}^-)$ et $P(\boldsymbol{y}^+)$ ne permettent pas l'obtention de la loi jointe $P(\boldsymbol{x}, \boldsymbol{y}; \mathbf{p}, \mathbf{p}_.)$. Une question se pose alors sur l'identifiabilité du modèle de transition proposé, où toutes les probabilités de transition $\mathbf{p}_.$ seraient à estimer. Dans la section suivante, nous montrons

que le modèle de transition complet n'est pas identifiable en paramètre.

3.2.3 Non identifiabilité du modèle de transition

L'objectif de ce travail est d'utiliser la variable $\boldsymbol{x}$ pour générer de nouvelles observations $\hat{\boldsymbol{y}}$. Dans la section précédente, nous proposons d'utiliser la loi jointe $P(\boldsymbol{x},\boldsymbol{y};\mathbf{p},\mathbf{p}_.)$. Néanmoins, le contexte de ce travail implique une absence totale de connaissances de la loi $P(\boldsymbol{y}|\boldsymbol{x};\mathbf{p}_.)$, dû aux données manquantes du problème. D'autre part, l'unique utilisation des lois marginales $P(\boldsymbol{x})$ et $P(\boldsymbol{y})$ n'apporte pas la quantité d'informations nécessaire à l'obtention de la loi jointe $P(\boldsymbol{x},\boldsymbol{y};\mathbf{p},\mathbf{p}_.)$. Cette faible quantité d'information disponible implique la non identifiabilité en paramètre du modèle de transition correspondant à la loi $P(\boldsymbol{y}|\boldsymbol{x};\mathbf{p}_.)$. Ce problème peut se formaliser sous la forme d'un problème probabiliste. En effet, la formule des probabilités totales, issue du cadre probabiliste, implique la relation suivante :

$$P(\boldsymbol{y};\mathbf{p}_.,\mathbf{p}) = \sum_{h=1}^{p} P(\boldsymbol{y}|(\boldsymbol{x})_h;\mathbf{p}_.)P((\boldsymbol{x})_h;\mathbf{p}). \tag{3.4}$$

En supposant le paramètre $\mathbf{p}$ connu (ce qui est réaliste car les observations $\mathbf{x}^-$ sont supposés disponibles en quantité suffisante pour bien estimer $\mathbf{p}$), le modèle de transition défini par l'équation 3.4 est dit "identifiable en paramètre" s'il respecte la définition suivante :

Définition : Avec $\mathbf{p}$ connu, un modèle $P(\boldsymbol{y};\mathbf{p}_.,\mathbf{p})$ est dit identifiable en $\mathbf{p}_.$ si pour un modèle δ donné, il n'existe qu'un seul vecteur de paramètres $\mathbf{p}_.$ possible. Soit, $\{\forall \mathbf{p}_., \mathbf{p}_.' \in \mathcal{P}., P(.;\mathbf{p}_.,\delta) = P(.;\mathbf{p}_.',\delta) \implies \mathbf{p}_. = \mathbf{p}_.'\}$, où $\mathcal{P}.$ est l'espace des probabilités $\mathbf{p}_.$

Le modèle de transition défini actuellement, correspondant à l'équation 3.4, suppose l'estimation de l'ensemble des paramètres $\mathbf{p}_.$. Or, la faible quantité d'informations disponibles pour le modèle de transition implique plusieurs vecteurs de paramètres $\mathbf{p}_.$ possibles pour un modèle donné. L'exemple suivant illustre la non identifiabilité du modèle de transition.

Exemple : Dans cet exemple, représenté par la figure 3.2 le modèle supposé est relativement simple. Dans cet exemple, $p = 2$ et $q = 3$. La formule des probabilités totales, issue du cadre probabiliste, implique la relation 3.4. L'obtention des probabilités de transition $P(\boldsymbol{y}|\boldsymbol{x})$ revient alors à un système linéaire à 6 inconnues, correspondant aux

paramètres $p_{hh'}$, tel que :

$$\begin{cases} p_1 p_{11} + p_2 p_{21} = q_1 \\ p_1 p_{12} + p_2 p_{22} = q_2 \\ p_1 p_{13} + p_2 p_{22} = q_3 \end{cases}$$

avec $q_{h'} = \sum_{h=1}^{p} p_h p'_{hh'} \; \forall h'$. Dans ce système, les paramètres p_h et $q_{h'}$ sont supposés connus, seul les paramètres $p_{hh'}$ sont inconnus.

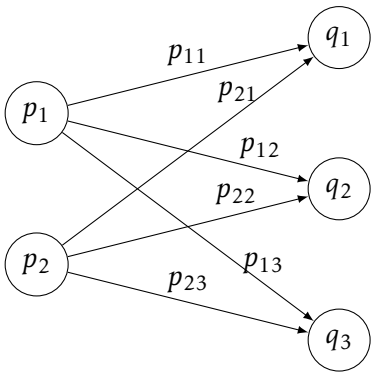


FIGURE 3.2 – Exemple de graphe représentant un modèle avec $p = 2$ et $q = 3$

Malgré les contraintes liées aux probabilités, soit : $\sum_{h'=1}^{q} p_{hh'} = 1 \; \forall h$, ce système ne comporte pas de solution unique. Le nombre d'inconnues est trop important par rapport à l'information disponible. En effet, si on pose le système linéaire sous forme matricielle, tel que présenté en Annexe A, les données disponibles ne permettent pas l'obtention d'un matrice de plein rang nécessaire à l'unicité de la solution du système. En effet, ce modèle implique que le rang de la matrice d'information soit inférieur au rang du vecteur d'inconnu. Selon le théorème de Rouché-Fonténé [102], présenté en Annexe A, cela implique que le système à une infinité de solutions.

Exemple numérique : Si l'on pose $\mathbf{p} = (0.6, 0.4)$ et en notant $y_i \sim M_q(\mathbf{q})$ où $\mathbf{q} = (0.5, 0.3, 0.2)$, les solutions possibles dépendent des valeurs de p_{21} et p_{23}. Cependant, ces paramètres peuvent prendre n'importe qu'elle valeur sur l'intervalle [0,1]. De ce fait, si l'on fixe $p_{21} = 0.5$ et $p_{23} = 0.2$, la solution de ce système est $\mathbf{p}_1 = (0.5, 0.3, 0.2)$ et $\mathbf{p}_2 = (0.5, 0.3, 0.2)$. Alors que si l'on fixe $p_{21} = 0.2$ et $p_{23} = 0.4$, on obtient $\mathbf{p}_1 = (0.7, 0.23, 0.07)$ et $\mathbf{p}_2 = (0.2, 0.4, 0.4)$ qui est aussi solution du système. Plusieurs vecteurs $\mathbf{p}_.$ sont donc possibles pour un même modèle, ce qui implique que le modèle n'est pas identifiable en paramètres. Afin de le rendre identifiable, nous proposons de le contraindre en réduisant le nombre de probabilités de transition $\mathbf{p}_.$ à estimer.

3.3 Proposition de contraintes d'identifiabilité

3.3.1 Contraintes d'interprétation de type binaire

Dans la section précédente, il a été montré que le modèle complet proposé n'est pas identifiable en paramètres, le nombre de paramètres inconnus étant trop important (6 dans l'exemple précédent), et l'information disponible insuffisante. Afin de restreindre le nombre de paramètres inconnus et rendre le modèle identifiable, nous proposons de le contraindre. Différents types de contraintes peuvent être proposés. Néanmoins, il est nécessaire que les contraintes appliquées soient simples et interprétables dans le modèle final. En effet, l'objectif d'interprétabilité du modèle énoncé dans la section 1.2, requiert de ne pas polluer le modèle avec des contraintes non interprétables. Pour répondre à ces attentes, nous proposons une contrainte très simple, de type binaire, qui consiste à fixer certaines valeurs de probabilités de transition $\mathbf{p}_{.}$ à zéro. De cette manière, le nombre de paramètres à estimer est réduit et il est aisé d'interpréter cette contrainte dans le modèle. Par exemple, pour un modèle où $p = 2$ et $q = 3$, le nombre de probabilités de transition total est de 8, dont 5 sont des probabilités à estimer et 3 sont des paramètres non libres. Avec les contraintes de type binaire, seul 3 paramètres sont à estimer. De même pour un modèle où $p = 3$ et $q = 5$, le modèle complet requiert 12 paramètres à estimer alors que le modèle sous contrainte n'en nécessite que 6. De plus, le cas d'usage étudié montre que ce type de contrainte est également intuitif dans un cadre d'application réelle. En effet, en reprenant le cas d'usage étudié, il semble peu probable qu'un internaute ayant choisi du tiers avant la modification, choisisse du Tous Risque ++ après la modification, à cause de la modification. Supposer que certaines probabilités de transition peuvent être fixées à zéro ne parait alors pas incohérent.

Formalisation des contraintes

Un modèle doté de ces contraintes de type binaire est alors noté $\boldsymbol{\delta} = \{\boldsymbol{\delta}_h\}_{h=1,\dots,p}$ avec $\boldsymbol{\delta}_h = (\delta_{hh'})_{h=1,\dots,q}$ où $\delta_{hh'} = 0$ si $p_{hh'} = 0$, et $\delta_{hh'} = 1$ sinon. Il est également à noter que la somme $\sum_{h'=1}^{q} p_{hh'} = 1 \ \forall h$. Un modèle peut alors être modélisé par la matrice Δ, une matrice diagonale de taille $pq \times pq$, indiquant les probabilités de transition à estimer ou non, et est de la forme :

$$\Delta = \begin{pmatrix} 1-\delta_{11} & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 1-\delta_{hh'} & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 1-\delta_{pq} \end{pmatrix}.$$

L'application de ces contraintes amène alors à un ensemble de modèles composés de paramètres à estimer et de paramètres contraints à zéro. La figure 3.3 montre trois exemples de modèles correspondant à ce type de contraintes où les arcs orientés représentés correspondent aux probabilités de transition estimées. Dans ces modèles, les contraintes se visualisent par l'absence d'arc entre les modalités des différentes variables. Ce type de contraintes permet de réduire le nombre de paramètres à estimer tout en gardant un modèle interprétable. Dans la section suivante, nous proposons des contraintes supplémentaires afin de s'assurer de l'identifiabilité en paramètres des différents modèles contraints.

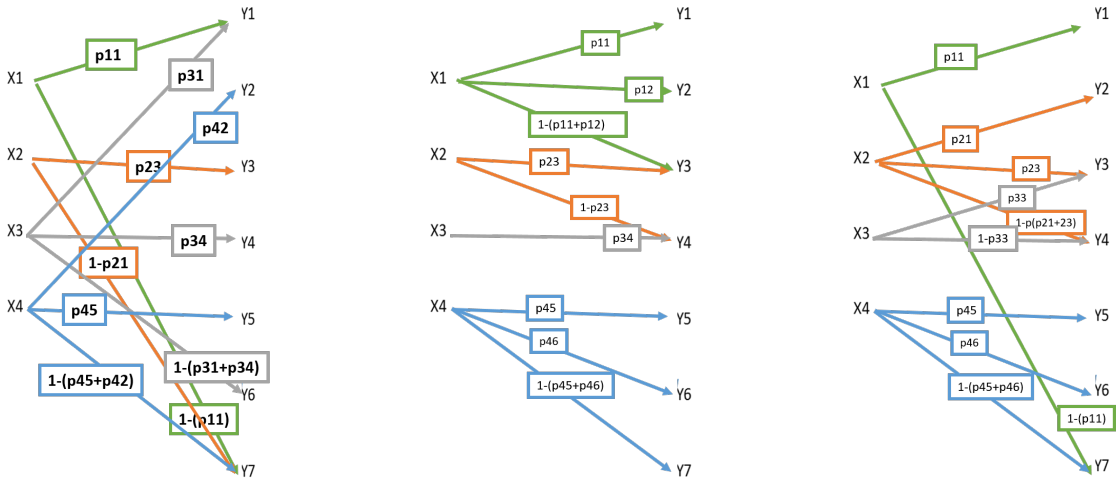


FIGURE 3.3 – Exemples de 3 modèles comportant les contraintes de type binaire, où l'absence d'arcs indique les probabilités de transition fixées à zéro.

3.3.2 Contraintes d'identifiabilité

Les contraintes de types binaires définies dans la section précédente peuvent entraîner des modèles ne respectant pas les données initiales du problème et ne suffisent pas à garantir l'identifiabilité des modèles contraints. Les données initiales du problème

indiquent le nombre de modalités p et q des variables $\boldsymbol{x}$ et $\boldsymbol{y}$. Chaque modèle contraint doit alors comporter ce même nombre de modalités p et q, ce qui implique que les probabilités de transition fixées à zéro ne soient pas concentrées sur une même modalité de $\boldsymbol{x}$ ou de $\boldsymbol{y}$. Chaque modèle contraint doit alors également respecter la contrainte supplémentaire suivante : il existe au moins une probabilité de transition non nulle entre chacune des p modalités de $\boldsymbol{x}$ et q modalités de $\boldsymbol{y}$.

Ce qui implique :

$$\begin{cases} \sum_{h'=1}^{q}(\delta_{hh'}) \geq 1, \forall h \\ \sum_{h=1}^{p}(\delta_{hh'}) \geq 1, \forall h'. \end{cases}$$

Ces contraintes formalisent l'ensemble des modèles possibles selon les probabilités de transition fixées à zéro. Un modèle est identifiable en paramètres s'il répond à la définition indiquée dans la section 3.2.3 :

Condition nécessaire Une condition nécessaire, détaillée en Annexe A, à l'obtention d'un modèle identifiable est que le nombre de probabilités fixées à zéro soit compris dans les bornes suivantes :

$$m - (q + \dim(p)) \leq \sum_{h=1}^{q}\sum_{i=1}^{p}(1 - \delta_{hh'}) \leq m - \max(p, q) \tag{3.5}$$

où $m = pq$ correspond à la taille du vecteur $\mathbf{p}_{.}$.

Condition suffisante De plus, une condition suffisante à l'obtention d'un modèle identifiable, également détaillée en Annexe A, est que le nombre de probabilités de transition fixées à zéro soit supérieur à la quantité suivante :

$$\sum_{h'=1}^{q}\sum_{h=1}^{p}(1 - \delta_{hh'}) \geq m - (q + \dim(p)) \,. \tag{3.6}$$

L'identifiabilité du modèle nécessaire à l'estimation des paramètres implique donc de poser des contraintes sur celui-ci. Nous avons proposé la contrainte très simple de fixer un certain nombre de probabilités de transition à zéro afin que celles-ci soient interprétables facilement. L'identifiabilité des modèles garantissant une solution unique pour chaque vecteur de paramètres, il est désormais possible d'estimer les probabilités

de transition non contraintes en garantissant leur unicité et interprétabilité. La partie suivante présente la méthode d'estimation utilisée, son application à notre problème et ses performances.

3.4 Estimation des paramètres

> La vraisemblance ne se confond pas avec la vérité, ni le réel avec sa représentation.
>
> *Grégoire Bouillier*

L'estimation des paramètres d'un modèle peut être réalisée par diverses méthodes. En statistique, les deux méthodes les plus répandues sont : la méthode des moments et les méthodes du maximum de vraisemblance. Ces deux méthodes sont réputées pour leurs propriétés statistiques et leur simplicité de mise en oeuvre.

La méthode des moments proposée par [85] puis étendue par [50] repose sur la loi des grands nombres. Elle consiste à estimer de façon empirique l'espérance ou même la variance d'un échantillon. Par exemple, pour une variable aléatoire multinomiale $\boldsymbol{x}$ où $\boldsymbol{x}_i \sim M_p(\mathbf{p})$, avec $\mathbf{p} = (p_h)_{h=1,\dots,p}$, l'espérance de $\boldsymbol{x}$ est $\mathbb{E}[(\boldsymbol{x})_h] = np_h$, l'estimateur $\hat{p}_h$ de p_h par la méthode des moments revient alors à la proportion de n_h dans l'échantillon. Cette méthode est généralement utilisée pour sa simplicité de mise en oeuvre et sa rapidité.

Cependant, le cadre de notre travail implique de travailler avec des échantillons contenant des données manquantes. Dans ce cadre, c'est généralement l'estimateur du maximum de vraisemblance qui est privilégié, notamment pour ses propriétés asymptotiques. La simplicité de mise en oeuvre de l'estimateur du maximum de vraisemblance, ses propriétés statistiques et sa pertinence pour l'estimation de paramètres en présence de données manquantes font que nous choisissons d'utiliser cet estimateur dans ce travail.

3.4.1 Maximisation de la vraisemblance

La méthode du maximum de vraisemblance (MV) introduite par [37] est, en pratique, la méthode la plus utilisée pour l'estimation de paramètres. Dans cette méthode, les observations $\mathbf{x} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$ sont supposées être des réalisations i.i.d d'une variable aléatoire $\boldsymbol{X}$ de distribution $f(.;\boldsymbol{\theta})$ dans le cas continu et $P(.;\boldsymbol{\theta})$ dans le cas discret. On appelle **fonction de vraisemblance** pour l'échantillon $\mathbf{x}$, la fonction de paramètres $\boldsymbol{\theta}$ telle que :

$$L(\boldsymbol{\theta};\mathbf{x}) = \begin{cases} P(\boldsymbol{X}_1 = \boldsymbol{x}_1,\ldots,\boldsymbol{X}_n = \boldsymbol{x}_n;\boldsymbol{\theta}) \text{ si les variables aléatoires } \boldsymbol{X}_i \text{ sont discrètes,} \\ f(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n;\boldsymbol{\theta}) \text{ si les variables aléatoires } \boldsymbol{X}_i \text{ sont continues.} \end{cases}$$

Lorsque toutes les observations sont de mêmes lois et indépendantes, la vraisemblance des paramètres s'écrit :

$$L(\boldsymbol{\theta};\mathbf{x}) = \begin{cases} \prod_{i=1}^{n} P(\boldsymbol{X}_i = \boldsymbol{x}_i;\boldsymbol{\theta}) \text{ si les variables aléatoires } \boldsymbol{X}_i \text{ sont discrètes,} \\ \prod_{i=1}^{n} f(\boldsymbol{x}_i;\boldsymbol{\theta}) \text{ si les variables aléatoires } \boldsymbol{X}_i \text{ sont continues.} \end{cases} \tag{3.7}$$

Le principe du MV est de maximiser la vraisemblance des paramètres de l'échantillon $\mathbf{x}$ par la fonction $L(\boldsymbol{\theta};\mathbf{x})$, où le produit des fonctions de densité des individus est donné par l'indépendance entre les individus. Le MV repose sur l'idée que le meilleur estimateur de $\boldsymbol{\theta}$, c'est à dire la valeur de $\boldsymbol{\theta}$ correspondant le mieux aux données, est la valeur qui maximise la vraisemblance, définie par l'équation 3.7, notée $\hat{\boldsymbol{\theta}}$. Les variables étudiées dans ce travail étant des loi multinomiales, nous nous focalisons sur le cas discret, où l'estimateur (EMV) s'écrit :

$$\hat{\boldsymbol{\theta}} = \operatorname{argmax} \prod_{i=1}^{n} P(\boldsymbol{X}_i = \boldsymbol{x}_i;\boldsymbol{\theta}). \tag{3.8}$$

Généralement, c'est la log-vraisemblance qui est utilisée, donnée par :

$$\ell(\boldsymbol{\theta};\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) = \ln(L((\boldsymbol{\theta};\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)) = \sum_{i=1}^{n} \ln(P(\boldsymbol{X}_i = \boldsymbol{x}_i;\boldsymbol{\theta})). \tag{3.9}$$

Sous les conditions de régularité standards, l'estimateur du maximum de vraisemblance est asymptotiquement optimal puisqu'il atteint la borne de Cramér-Rao [39]. Pour l'estimation des paramètres, la méthode du maximum de vraisemblance est généralement préférée pour ses propriétés asymptotiques, qui sont :

— Obtention d'estimateurs non biaisés,
— Obtention d'estimateurs optimaux par atteinte de la borne de Cramer-Rao,
— Normalité.

Application à notre cas

L'estimation par maximum de vraisemblance consiste à maximiser la vraisemblance des paramètres conditionnellement aux données. Dans le cadre de notre travail, l'objectif est d'estimer la loi jointe $P(\boldsymbol{x},\boldsymbol{y}|\delta;\mathbf{p},\mathbf{p}_{\cdot})$. Dans la suite de ce travail, la notation du modèle δ est rendue implicite et la loi jointe est alors notée $P(\boldsymbol{x},\boldsymbol{y};\mathbf{p},\mathbf{p}_{\cdot})$. Dans la section 3.1.1, les partitions $\mathbf{x} = (\mathbf{x}^-,\mathbf{x}^+)$ et $\mathbf{y} = (\mathbf{y}^-,\mathbf{y}^+)$, correspondant aux données avant et après la modification ont été réalisées. Reprenant ces partitions, la loi jointe est alors réécrite $P(\boldsymbol{x}^-,\boldsymbol{x}^+,\boldsymbol{y}^-,\boldsymbol{y}^+)$. Les couples $(\boldsymbol{x}_i,\boldsymbol{y}_i)$ étant supposés stochastiquement indépendants, il vient $P(\boldsymbol{x}^-,\boldsymbol{x}^+,\boldsymbol{y}^-,\boldsymbol{y}^+) = P(\boldsymbol{x}^-,\boldsymbol{y}^-)P(\boldsymbol{x}^+,\boldsymbol{y}^+)$ où $\mathbf{x}^+$ et $\mathbf{y}^-$ sont des données manquantes. En effet, pour chaque couple $(\boldsymbol{x}_i,\boldsymbol{y}_i)$ seule une donnée partielle a été observée. La log-vraisemblance de la loi jointe pour les données observées est alors définie par l'équation 3.3. En notant $\boldsymbol{\theta} = (\mathbf{p},\mathbf{p}_{\cdot})$, nous avons alors :

$$\hat{\boldsymbol{\theta}} = \underset{\theta\in\boldsymbol{\Theta}}{\operatorname{argmax}}\ \ell(\boldsymbol{\theta};\mathbf{x}^-,\mathbf{y}^+). \tag{3.10}$$

Il est courant que l'estimation des paramètres et la maximisation de la log-vraisemblance ne soit pas explicite, notamment lorsque les échantillons contiennent des données manquantes. Divers algorithmes d'estimations tel que celui de Newton-Raphson peuvent alors être utilisés pour réaliser la maximisation et l'estimation des paramètres. En présence de données manquantes, cependant, l'algorithme privilégié est l'algorithme Expectation Maximisation (EM) [78]. L'algorithme EM, présenté par Dempster, Laird et Rupin [30] est un algorithme itératif permettant de maximiser la log-vraisemblance, justement, en présence de données manquantes. Cet algorithme est maintenant présenté plus en détail.

3.4.2 Algorithme Expectation Maximisation (EM)

L'algorithme Expectation Maximisation (EM), [78],[30], est un algorithme itératif dont le principe général est, dans le cadre d'un problème comportant des données manquantes tel que celui étudié, que pour chaque itération, les données observées sont complétées avec les données non observées. Le but final étant de rendre l'optimisation aussi aisée qu'en ayant les données complètes disponibles. Cela rend cet algorithme particulièrement adapté au problème avec des données manquantes.

Principe L'algorithme EM part du principe qu'il est généralement plus simple d'optimiser la vraisemblance des données complètes, que la vraisemblance avec des données incomplètes. L'algorithme consiste alors en la reconstitution des

données manquantes afin de maximiser la vraisemblance des données complètes. Pour cela, l'algorithme fonctionne de manière récursive en deux étapes.

Expectation La première étape est l'étape dite d'*Expectation (E)*. Elle consiste à calculer $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(r)})$, correspondant à l'espérance conditionnelle de la log-vraisemblance des données complètes (connues et inconnues) sachant les données observées et les paramètres courants $\boldsymbol{\theta}^{(r)}$.

Maximisation La seconde étape est l'étape dite de *Maximisation (M)*. Elle consiste à maximiser $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(r)})$ en $\boldsymbol{\theta}$. Ce qui revient à construire une suite vérifiant :

$$\boldsymbol{\theta}^{(r+1)} = \underset{\theta \in \boldsymbol{\Theta}}{\operatorname{argmax}}\, Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(r)}). \tag{3.11}$$

Outre le fait qu'il soit particulièrement adapté aux problèmes ayant des données manquantes, l'algorithme EM possède également les propriétés suivantes :

— La vraisemblance croît à chaque étape,

— Les contraintes sont naturellement vérifiées,

— Il est peu coûteux en mémoire.

Comparé à l'algorithme de Newton, l'algorithme EM converge lentement. Cela vient du fait que l'algorithme de Newton a une convergence quadratique contrairement à l'algorithme EM qui a une convergence linéaire. Cependant, l'algorithme de Newton est plus complexe à mettre en place et n'assure pas un résultat convergeant si la fonction n'est pas convexe.

3.4.3 Estimation par maximum de vraisemblance profilée

Dans un premier temps, une estimation par vraisemblance profilée [23], [80] a été réalisée. La vraisemblance profilée permet de «concentrer» la fonction de log-vraisemblance sur le paramètre d'intérêt $\mathbf{p}_{.}$, après avoir éliminé le paramètre parasite $\mathbf{p}$. Le nombre de données $\mathbf{x}^{-}$ étant suffisant, il nous permet d'avoir un bon estimateur $\mathbf{p}$. A l'inverse, n'ayant aucune information sur les paramètres $\mathbf{p}_{.}$, notre intérêt se porte donc particulièrement sur ces paramètres. Dans ce travail, l'objectif est d'estimer les paramètres $\mathbf{p}$ et $\mathbf{p}_{.}$ du modèle $P(\boldsymbol{x}^{-}, \boldsymbol{y}^{+}; \mathbf{p}, \mathbf{p}_{.})$. Sous les hypothèses d'indépendances des couples d'observations $(\boldsymbol{x}_i^{-}, \boldsymbol{y}_i^{-})$ et $(\boldsymbol{x}_i^{+}, \boldsymbol{y}_i^{+})$, ce modèle peut s'écrire sous la forme $P(\boldsymbol{x}^{-}; \mathbf{p})P(\boldsymbol{y}^{+}; \mathbf{p}, \mathbf{p}_{.})$. C'est cette forme qui est utilisée pour l'estimation des paramètres par vraisemblance profilée.

Définition : La vraisemblance profilée consiste à maximiser la vraisemblance en deux étapes. D'abord par rapport aux paramètres $\mathbf{p}$ avec $L(\mathbf{p}; \mathbf{x}^{-})$, puis par rapport

aux paramètres d'intérêts $\mathbf{p}_{.}$ avec $L(\mathbf{p}_{.}, \hat{\mathbf{p}}; \mathbf{x}^{-}, \mathbf{y}^{+})$, où $\hat{\mathbf{p}}$ est l'estimateur obtenu par $L(\mathbf{p}; \mathbf{x}^{-})$.

Estimateur $\hat{\mathbf{p}}$: Utilisant les données observées $\mathbf{x}^{-}$, l'estimateur $\hat{\mathbf{p}}$ est aisément obtenue par l'EMV. En effet, comme $\boldsymbol{x}_i \sim M_p(\mathbf{p})$, la log-vraisemblance de l'estimateur s'écrit $\ell(\mathbf{p}; \mathbf{x}^{-}) = \sum_i^{n^-} \sum_h^p x_{ih}^{-} \ln p_h$. Les propriétés de l'EMV présentées en section 3.4.1, notamment de convergence, s'appliquent alors à l'estimateur $\hat{\mathbf{p}} = \operatorname{argmax} \ell(\mathbf{p}; \mathbf{x}^{-})$.

Estimateur $\hat{\mathbf{p}}_{.}$: L'estimation des paramètres $\mathbf{p}_{.}$ nécessite la prise en compte des données manquantes $\mathbf{x}^{+}$ et $\mathbf{y}^{-}$. Le calcul de l'EMV requiert alors de passer par un algorithme de type EM, prenant en compte les données manquantes. L'estimation des paramètres $\mathbf{p}_{.}$ sera alors réalisée avec l'algorithme EM présenté en section 3.4.2. Utilisant le paramètre $\hat{\mathbf{p}}$ estimé en première étape, la vraisemblance des données complètes est la suivante :

$$L_c(\hat{\mathbf{p}}, \mathbf{p}_{.}; \mathbf{x}^{-}, \mathbf{x}^{+}, \mathbf{y}^{-}, \mathbf{y}^{+}) = \prod_{i=1}^{n^-} \prod_{h=1}^{p} \prod_{h'=1}^{q} (p_{hh'} \hat{p}_h)^{x_{ih}^{-} y_{ih'}^{-}} \prod_{i=1}^{n^+} \prod_{h=1}^{p} \prod_{h'=1}^{q} (p_{hh'} \hat{p}_h)^{x_{ih}^{+} y_{ih'}^{+}}. \tag{3.12}$$

Étape E : Expectation L'utilisation de la log-vraisemblance des données complétées permet l'obtention de l'étape d'estimation de l'algorithme par :
$Q(\mathbf{p}_{.} | \mathbf{p}_{.}^{(r)}) = \mathbb{E}[\ell_c(\mathbf{p}_{.}, \hat{\mathbf{p}}; \mathbf{x}^{-}, \mathbf{y}^{+}, \mathbf{x}^{+}, \mathbf{y}^{-}) | \mathbf{x}^{-}, \mathbf{y}^{+}; \hat{\mathbf{p}}, \mathbf{p}_{.}^{(r)}]$. L'espérance de la log-vraisemblance des données complétées $\ell_c(\mathbf{p}_{.}, \hat{\mathbf{p}}; \mathbf{x}^{-}, \mathbf{y}^{+}, \mathbf{x}^{+}, \mathbf{y}^{-})$ par rapport aux données observées et aux paramètres courants revient à calculer $Q(\mathbf{p}_{.} | \mathbf{p}_{.}^{(r)})$ en utilisant le fait que les données inconnues soient les données $(\mathbf{x}^{+}, \mathbf{y}^{-})$, et que les paramètres $\hat{\mathbf{p}}$ soit connus, ce qui revient à calculer :

$$\begin{aligned} Q(\mathbf{p}_{.}, \mathbf{p}_{.}^{(r)}) &= \sum_{i=1}^{n^-} \sum_{h=1}^{p} x_{ih}^{-} \ln p_h + \sum_{i=1}^{n^+} \sum_{h=1}^{p} \mathbb{E}[x_{ih}^{+} | y_{ih'}^{+}; \mathbf{p}_{.}^{(r)}, \hat{\mathbf{p}}] \ln p_h \\ &+ \sum_{i=1}^{n^-} \sum_{h=1}^{p} \sum_{h'=1}^{q} x_{ih}^{-} \mathbb{E}[y_{ih'}^{-} | x_{ih}^{-}; \mathbf{p}_{.}^{(r)}, \hat{\mathbf{p}}] \ln p_{hh'} + \sum_{i'=1}^{n^+} \sum_{h=1}^{p} \sum_{h'=1}^{q} \mathbb{E}[x_{ih}^{+} | y_{ih'}^{+}; \mathbf{p}_{.}^{(r)}, \hat{\mathbf{p}}] y_{ih'}^{+} \ln p_{hh'}. \end{aligned} \tag{3.13}$$

L'équation 3.13 fait alors apparaître $\mathbb{E}[x_{ih}^{+} | y_{ih'}^{+}; \mathbf{p}_{.}^{(r)}, \hat{\mathbf{p}}]$. Le calcul de cette espérance revient à calculer la probabilité a posteriori $P(x_{ih}^{+} = 1 | y_{ih'}^{+} = 1; \mathbf{p}_{.}^{(r)}, \hat{\mathbf{p}})$, par l'expression suivante :

$$P(x_{ih}^{+} = 1 | y_{ih'}^{+} = 1; \mathbf{p}_{.}^{(r)}, \hat{\mathbf{p}}) = \frac{P(y_{ih'}^{+} | x_{ih}^{+} = 1; p_{hh'}^{(r)}) P(x_{ih}^{+}; \hat{p}_h)}{\sum_{h=1}^{p} (P(y_{ih'}^{+} | x_{ih}^{+} = 1; p_{hh'}^{(r)}) P(x_{ih}^{+}; \hat{p}_h))}. \tag{3.14}$$

Par la suite, cette probabilité servant de poids (weight en anglais) dans les formules suivantes sera notée $w_{ih}^{(r)}$. D'autre part, l'équation 3.13 fait également apparaître $\mathbb{E}[y_{ih'}^{-}|x_{ih}^{-};\mathbf{p}_{.}^{(r)},\hat{\mathbf{p}}]$. Le calcul de cette espérance revient quant à lui, à calculer la probabilité $P(\boldsymbol{y}^{-}|\boldsymbol{x}^{-})$, soit $P(y_{ih'}^{-}|x_{ih}^{-}=1;p_{hh'}^{(r)})$. Par la suite, nous noterons cette probabilité $w_{ihh'}^{(r)}$.

L'équation 3.13 est finalement donnée par :

$$Q(\mathbf{p}_{.},\mathbf{p}_{.}^{(r)}) = \sum_{i=1}^{n^{-}}\sum_{h=1}^{p} x_{ih}^{-}\ln p_h + \sum_{i=1}^{n^{+}}\sum_{h=1}^{p} w_{ih}^{(r)}\ln p_h + \sum_{i=1}^{n^{-}}\sum_{h=1}^{p}\sum_{h'=1}^{q} x_{ih}^{-}w_{ihh'}^{(r)}\ln p_{hh'} + \sum_{i'=1}^{n^{+}}\sum_{h=1}^{p}\sum_{h'=1}^{q} w_{ih}^{(r)}y_{ih'}^{+}\ln p_{hh'}. \quad (3.15)$$

L'étape de maximisation consiste alors à maximiser $Q(\mathbf{p}_{.},\mathbf{p}_{.}^{(r)})$ en $\mathbf{p}_{.}$, donnée par l'équation 3.15.

Etape M : Maximisation L'actualisation des paramètres $p_{hh'}^{(r+1)}$ s'obtient aisément par :

$$p_{hh'}^{(r+1)} = \frac{1}{n}\sum_{i=1}^{n^{-}}(x_{ih}^{-}\mathbb{E}[y_{ih'}^{-}|x_{ih}^{-};\mathbf{p}_{.}^{(r+1)},\hat{\mathbf{p}}]) + \sum_{i=1}^{n^{+}}\mathbb{E}[x_{ih}^{+}|y_{ih'}^{+};\mathbf{p}_{.}^{(r+1)},\hat{\mathbf{p}}]y_{ih'}^{+} \quad (3.16)$$

soit

$$p_{hh'}^{(r+1)} = \frac{1}{n}\sum_{i=1}^{n^{-}}(x_{ih}^{-}w_{ihh'}^{(r+1)}) + \sum_{i=1}^{n^{+}} w_{ih}^{(r+1)}y_{ih'}^{+}. \quad (3.17)$$

Bien que le terme "vraisemblance" apparaisse pour cette méthode d'estimation, les propriétés de l'EMV pour l'estimateur global ne sont pas totalement garanties pour cette méthode d'estimation. Néanmoins, dans la partie 3.3.2 et l'annexe A, il a été montré que sous l'hypothèse que les paramètres $\mathbf{p}$ soient connus, le vecteur de paramètres $\mathbf{p}_{.}$ est identifiable et donc la solution est unique. D'autre part, les propriétés de l'EMV, calculé pour les paramètres $\mathbf{p}$, impliquent que les estimateurs $\hat{\mathbf{p}}$ convergent vers $\mathbf{p}$. Par le théorème de l'application continue [75] (continuous mapping theorem en anglais), il est alors aisé d'obtenir la convergence de l'estimateur $\hat{\mathbf{p}}_{.}$.

Dans le calcul de la vraisemblance profilée, il est considéré que l'estimation des paramètres $\mathbf{p}$ ne dépend que des données $\mathbf{x}^{-}$, or le calcul de la loi $P(\boldsymbol{y}^{+};\mathbf{p},\mathbf{p}_{.})$ implique que l'estimation des paramètres $\mathbf{p}$ dépendent aussi des données $\mathbf{y}^{+}$. Une méthode d'estimation jointe est alors proposée.

3.4.4 Estimation jointe des paramètres par maximum de vraisemblance

L'objectif de cette section est d'estimer l'ensemble des paramètres $\boldsymbol{\theta} = (\mathbf{p}, \mathbf{p}_{\cdot})$ du modèle $P(\boldsymbol{x}, \boldsymbol{y}; \mathbf{p}_{\cdot}, \mathbf{p})$. Contrairement à l'hypothèse réalisée pour l'estimation par vraisemblance profilée, l'estimation des paramètres $\mathbf{p}$ faisant partie du modèle $P(\boldsymbol{y}; \mathbf{p}_{\cdot}, \mathbf{p})$, ne dépend pas uniquement des données $\mathbf{x}^-$ mais également des données $\mathbf{y}^+$. Étant en présence de données manquantes, de nouveau, un algorithme EM est utilisé pour estimer l'ensemble des paramètres $\boldsymbol{\theta}$. Les variables utilisées suivant des lois multinomiales, et utilisant les propriétés d'indépendances des couples $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ énoncées précédemment, la vraisemblance complétée par les données manquantes $\mathbf{x}^+$, $\mathbf{y}^-$ est donnée par :

$$L_c(\mathbf{p}, \mathbf{p}_{\cdot}; \mathbf{x}^-, \mathbf{x}^+, \mathbf{y}^-, \mathbf{y}^+) = \prod_{i=1}^{n^-} \prod_{h=1}^{p} \prod_{h'=1}^{q} (p_{hh'} p_h)^{x_{ih}^- y_{ih'}^-} \prod_{i=1}^{n^+} \prod_{h=1}^{p} \prod_{h'=1}^{q} (p_{hh'} p_h)^{x_{ih}^+ y_{ih'}^+} \tag{3.18}$$

et la log-vraisemblance des données complètes se définie par :

$$\begin{aligned} \ell_c(\mathbf{p}_{\cdot}, \mathbf{p}; \mathbf{x}^-, \mathbf{y}^+, \mathbf{x}^+, \mathbf{y}^-) = & \sum_{i=1}^{n^-} \sum_{h=1}^{p} \sum_{h'=1}^{q} \underbrace{x_{ih}^-}_{connues} \underbrace{y_{ih'}^-}_{inconnues} \ln p_{hh'} + \sum_{i=1}^{n^-} \sum_{h=1}^{p} \underbrace{x_{ih}^-}_{connues} \ln p_h \\ & + \sum_{i'=1}^{n^+} \sum_{h=1}^{p} \sum_{h'=1}^{q} \underbrace{x_{ih}^+}_{inconnues} \underbrace{y_{ih'}^+}_{connues} \ln p_{hh'} + \sum_{i'=1}^{n^+} \sum_{h=1}^{p} \underbrace{x_{ih}^+}_{inconnues} \ln p_h. \end{aligned} \tag{3.19}$$

Étape E : Expectation L'utilisation de la log-vraisemblance des données complétées permet l'obtention de l'étape d'estimation $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)}) = \mathbb{E}[\ell_c(\mathbf{p}_{\cdot}, \mathbf{p}; \mathbf{x}^-, \mathbf{y}^+, \mathbf{x}^+, \mathbf{y}^-)|\mathbf{x}^-, \mathbf{y}^+; \mathbf{p}^{(r)}, \mathbf{p}_{\cdot}^{(r)}]$, où l'ensemble des paramètres $\boldsymbol{\theta}$ de l'algorithme sont à estimer. Ce qui revient à calculer :

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(r)}) = & \sum_{i=1}^{n^-} \sum_{h=1}^{p} x_{ih}^- \ln p_h + \sum_{i=1}^{n^+} \sum_{h=1}^{p} \mathbb{E}[x_{ih}^+ | y_{ih'}^+; \boldsymbol{\theta}^{(r)}] \ln p_h \\ & + \sum_{i=1}^{n^-} \sum_{h=1}^{p} \sum_{h'=1}^{q} x_{ih}^- \mathbb{E}[y_{ih'}^- | x_{ih}^-; \boldsymbol{\theta}^{(r)}] \ln p_{hh'} + \sum_{i'=1}^{n^+} \sum_{h=1}^{p} \sum_{h'=1}^{q} \mathbb{E}[x_{ih}^+ | y_{ih'}^+; \boldsymbol{\theta}^{(r)}] y_{ih'}^+ \ln p_{hh'}. \end{aligned} \tag{3.20}$$

L'équation 3.20 fait alors apparaître $\mathbb{E}[x_{ih}^+ | y_{ih'}^+; \boldsymbol{\theta}^{(r)}]$. Ce qui revient à calculer la probabilité a posteriori donnée par l'équation 3.14, où les paramètres $\hat{\mathbf{p}}$ sont désormais

estimés par $\mathbf{p}^{(r)}$ tel que :

$$P(x_{ih}^{+}=1|y_{ih'}^{+}=1;\boldsymbol{\theta}^{(r)})=\frac{P(y_{ih'}^{+}|x_{ih}^{+}=1;p_{hh'}^{(r)})P(x_{ih}^{+};p_{h}^{(r)})}{\sum_{h=1}^{p}(P(y_{ih'}^{+}|x_{ih}^{+}=1;p_{hh'}^{(r)})P(x_{ih}^{+};p_{h}^{(r)}))}. \tag{3.21}$$

De nouveau, cette probabilité servant de poids (weight en anglais) dans les formules suivante sera notée $w_{ih}^{(r)}$. A l'instar de l'équation 3.14, le calcul de l'espérance $\mathbb{E}[y_{ih'}^{-}|x_{ih}^{-};\boldsymbol{\theta}^{(r)}]$ revient à calculer la probabilité $w_{ihh'}^{(r)}$. L'équation 3.20 revient finalement à l'équation 3.15, où les paramètres $\hat{\mathbf{p}}$ sont désormais les paramètres $\mathbf{p}^{(r)}$ estimés par l'algorithme et actualisés à chaque itération. Pour l'estimation jointe des paramètres, l'étape de maximisation consiste alors à maximiser $Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(r)})$ en $\boldsymbol{\theta}$, donnée par l'équation 3.20.

Etape M : Maximisation L'actualisation des paramètres $p_{h}^{(r+1)}$ s'obtient par :

$$p_{h}^{(r+1)}=\frac{1}{n}\sum_{h=1}^{p}\left[\sum_{i=1}^{n^{-}}x_{ih}^{-}+\sum_{i=1}^{n^{+}}\mathbb{E}[x_{ih}^{+}|y_{ih'}^{+};\boldsymbol{\theta}^{(r+1)}]\right] \tag{3.22}$$

soit :

$$p_{h}^{(r+1)}=\frac{1}{n}\sum_{h=1}^{p}\left[\sum_{i=1}^{n^{-}}x_{ih}^{-}+\sum_{i=1}^{n^{+}}w_{ih}^{(r+1)}\right] \tag{3.23}$$

et l'actualisation des paramètres $p_{hh'}^{(r+1)}$ s'obtient aisément en reprenant l'équation 3.17.

3.5 Comparaison des méthodes d'estimations

L'objectif de cette section est de comparer les deux méthodes d'estimations des paramètres selon un modèle donné et les données observées $(\mathbf{x}^{-},\mathbf{y}^{+})$. Dans le cas de données réelles, l'information nécessaire pour valider l'estimation n'est pas disponible. Afin de juger la qualité et la fiabilité de l'estimation des paramètres estimés par l'algorithme, pour les deux méthodes, plusieurs jeux de données, suivant différents modèles, ont été simulés.

3.5.1 Données

Les expérimentations sont réalisées selon 5 jeux de données simulés. Chacun des jeux de données correspond à un modèle dont les caractéristiques sont présentées dans le

tableau 3.1. Pour chaque modèle un échantillon de taille n a été généré avec la fonction Rmultinom du logiciel R.

δ	p	q	n	$\#\delta_{hh'}$
δ_1	2	3	5000	3
δ_2	3	4	5000	5
δ_3	3	5	5000	6
δ_4	4	7	5000	8
δ_5	2	3	5000	3

TABLEAU 3.1 – Caractéristiques des modèles simulés avec $\#\delta_{hh'}$ le nombre de paramètres à estimer pour le modèle

3.5.2 Protocole

Pour chacun des 5 modèles, les algorithmes d'estimation ont été lancés 100 fois. Chaque algorithme est arrêté après 500 itérations. La moyenne de chaque paramètre estimé est calculée sur les résultats des 100 algorithmes d'estimation effectués pour chacun des modèles.

3.5.3 Résultats

Le tableau 3.2a présente les différents modèles et leurs paramètres $\mathbf{p}$, $\mathbf{p}_h$ simulés. Le tableau 3.2b présente la moyenne par paramètres estimés $\hat{\mathbf{p}}_h$ par vraisemblance profilée pour chacun des modèles. Le tableau 3.2c présente la moyenne par paramètres estimés conjointement $\hat{\mathbf{p}}$, $\hat{\mathbf{p}}_h$ avec l'algorithme EM pour chacun des modèles.

Les tableaux 3.2b et 3.2c permettent de constater que les algorithmes d'estimation fournissent tout deux une estimation assez précise des paramètres et proche des paramètres simulés, quel que soit le modèle. On remarque également des résultats similaires pour les deux algorithmes d'estimation. Cette similarité des résultats s'explique par la quantité de données observées $\mathbf{y}^+$. En effet, la faible quantité de données $\mathbf{y}^+$ influence peu l'estimation des paramètres $\mathbf{p}$ contrairement aux données observées $\mathbf{x}^-$ qui sont en quantité importante. Ces deux tableaux permettent également de constater que les deux méthodes d'estimation permettent d'avoir une estimation précise et fiable des paramètres, malgré les nombreuses données manquantes. D'autre part, il a été montré que la méthode d'estimation par vraisemblance profilée permet l'obtention d'estimateurs $\hat{\mathbf{p}}$ et $\hat{\mathbf{p}}_.$ convergents. En ce qui concerne la convergence des paramètres $\hat{\boldsymbol{\theta}}$ avec l'estimation jointe, nous ne possédons pas de résultats théoriques. Néanmoins, empiriquement, les résultats obtenus par cette méthode semblent également convergents. En effet, les 100

	Paramètres simulés				
δ	$\mathbf{p}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$
δ_1	(0.6,0.4)	(0,0.9,0.1)	(0.7,0,0.3)	-	-
δ_2	(0.5,0.3,0.2)	(0,0.7,0,0.3)	(0.6,0,0,0.4)	(0,0,0.8,0.2)	-
δ_3	(0.5,0.3,0.2)	(0.1,0,0,0,0.9)	(0,0.3,0.7,0,0)	(0.2,0,0,0.8,0)	-
δ_4	(0.4,0.1,0.2,0.3)	(0.65,0,0.35,0,0,0,0)	(0,0.9,0,0,0,0,0.1)	(0,0,0,0.6,0,0.3,0)	(0,0,0,0,1,0,0)
δ_5	(0.6,0.4)	(0.7,0,0.3)	(0,0.8,0.2)	-	-

(a) Paramètres simulés pour chaque modèle

	paramètres moyens estimés par vraisemblance profilée				
δ	$\hat{\mathbf{p}}$	$\hat{\mathbf{p}}_1$	$\hat{\mathbf{p}}_2$	$\hat{\mathbf{p}}_3$	$\hat{\mathbf{p}}_4$
δ_1	(0.6,0.4)	(0,0.85,0.15)	(0.74,0,0.26)	-	-
δ_2	(0.5,0.3,0.2)	(0,0.68,0,0.32)	(0.63,0,0,0.37)	(0,0,0.78,0.22)	-
δ_3	(0.5,0.3,0.2)	(0.19,0,0,0,0.0.81)	(0,0.32,0.68,0,0)	(0.23,0,0,0.77,0)	-
δ_4	(0.4,0.1,0.2,0.3)	(0.66,0,0.34,0,0,0,0)	(0,0.88,0,0,0,0,0.12)	(0,0,0,0.67,0,0.33,0)	(0,0,0,0,0.98,0,0.02)
δ_5	(0.6,0.4)	(0.72,0,0.28)	(0,0.79,0.21)	-	-

(b) Paramètres estimés par vraisemblance profilée

	paramètres moyens estimés par vraisemblance jointe				
δ	$\hat{\mathbf{p}}$	$\hat{\mathbf{p}}_1$	$\hat{\mathbf{p}}_2$	$\hat{\mathbf{p}}_3$	$\hat{\mathbf{p}}_4$
δ_1	(0.61,0.39)	(0,0.87,0.13)	(0.0.72,0,0.28)	-	-
δ_2	(0.51,0.29,0.20)	(0,0.69,0,0.31)	(0.61,0,0,0.39)	(0,0,0.79,0.21)	-
δ_3	(0.52,0.28,0.20)	(0.19,0,0,0,0.0.81)	(0,0.32,0.68,0,0)	(0.23,0,0,0.77,0)	-
δ_4	(0.42,0.09,0.19,0.3)	(0.64,0,0.36,0,0,0,0)	(0,0.89,0,0,0,0,0.11)	(0,0,0,0.68,0,0.32,0)	(0,0,0,0,0.1,0,0)
δ_5	(0.59,0.41)	(0.71,0,0.29)	(0,0.80,0.20)	-	-

(c) Paramètres estimés par vraisemblance jointe

Tableau 3.2 – Estimation moyenne des paramètres $\mathbf{p}$, $\mathbf{p}_{\cdot}$ selon la méthode d'estimation utilisée

algorithmes EM lancés pour réaliser ces expérimentations dans le cadre de la vraisemblance jointe ont retournés des résultats similaires, ce que l'on peut constater avec le tableau 3.2c, et l'estimation moyenne très proche des paramètres simulés. D'autre part, la figure 3.4 permet également ce constat. En effet, la figure 3.4 montre l'évolution de l'estimation des paramètres $\hat{p}_{11}$ et $\hat{p}_{13}$ pour 100 algorithmes EM, où l'ensemble des paramètres est initialisé aléatoirement. Bien qu'il n'y ait pas tous les paramètres à estimer, cette figure permet de constater que ces deux paramètres convergent vers la même valeur pour chacun des algorithmes. Le détail de l'échantillon de données utilisé est énoncé en partie 3.6.1.

Bien que donnant des résultats similaires, dans la suite de ce travail, la méthode d'estimation jointe sera utilisée.

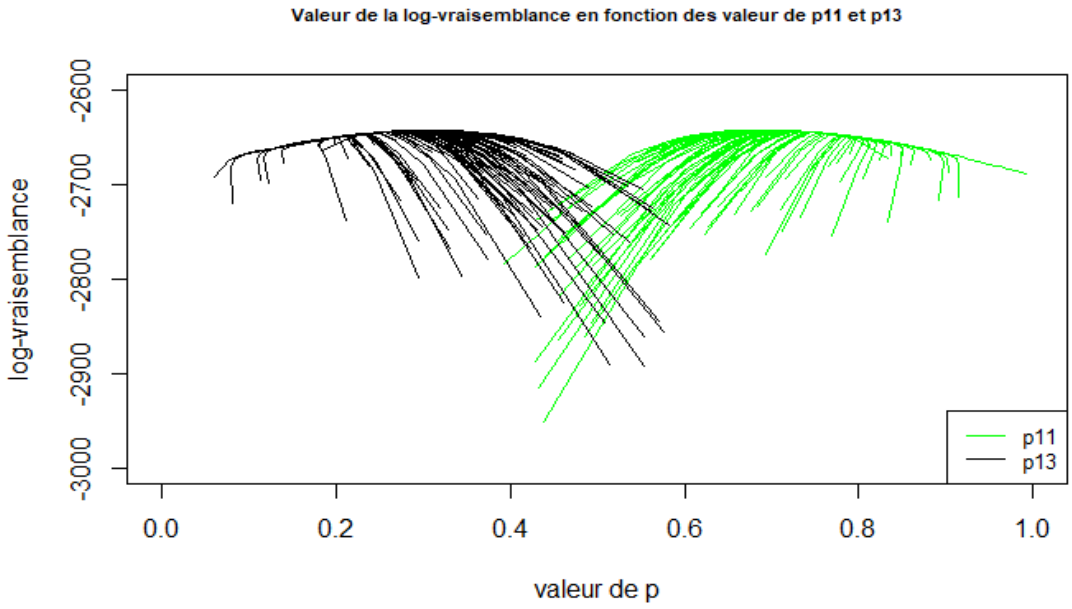


FIGURE 3.4 – Évolution de la log-vraisemblance de 100 EM selon l'évolution des estimateurs $\hat{p}_{11}$ et $\hat{p}_{13}$ au cours des algorithmes

3.6 Problématique autour de l'algorithme EM

3.6.1 Initialisation

Généralement, la vraisemblance comporte de nombreux maxima locaux, ce qui implique que le résultat obtenu dépend de l'initialisation de l'algorithme. Selon la stratégie d'initialisation utilisée, il peut être nécessaire de relancer plusieurs fois l'algorithme. Différentes stratégies d'initialisation existent, [12]. Pour notre algorithme, une stratégie d'initialisation aléatoire des paramètres θ peut être utilisée car il ne semble pas y avoir de maxima locaux. En effet, bien que les paramètres soient initialisés aléatoirement, la log-vraisemblance calculée par l'algorithme EM converge toujours vers la même valeur, comme le montre la figure 3.5. La figure 3.5 montre les résultats de la log-vraisemblance pour 100 algorithmes EM. Chaque algorithme a été lancé avec une initialisation aléatoire des paramètres $\mathbf{p}$ et $\mathbf{p}_.$ sur un même échantillon de données simulées, de taille $n = 300$ où $\mathbf{p} = (0.6, 0.4)$, $\mathbf{p}_1 = (0.7, 0, 0.3)$ et $\mathbf{p}_2 = (0, 0.8, 0.2)$. D'autre part, la figure 3.4 montre les résultats de 100 algorithmes EM en fonction de l'évolution de l'estimation des paramètres $\hat{p}_{11}$ et $\hat{p}_{13}$ au cours des différentes itérations de l'algorithme EM. Chaque algorithme a été lancé avec une initialisation aléatoire des paramètres $\mathbf{p}$ et $\mathbf{p}_.$ sur un même échantillon de

données simulées de taille $n = 1500$, où $\mathbf{p} = (0.6, 0.4)$, $\mathbf{p}_1 = (0.7, 0, 0.3)$ et $\mathbf{p}_2 = (0, 0.8, 0.2)$. On constate également que pour chaque algorithme, la log-vraisemblance atteint un maximum global lorsque le paramètre estimé $\hat{p}_{11} \approx 0.7$ et lorsque le paramètre estimé $\hat{p}_{13} \approx 0.3$, soit les valeurs simulées respectives des deux paramètres. Sur la figure 3.4, les courbes vertes indiquent les résultats de la log-vraisemblance selon la valeur du paramètre estimé $\hat{p}_{11}$ et les courbes noires indiquent les résultats de la log-vraisemblance selon la valeur du paramètre estimé $\hat{p}_{13}$.

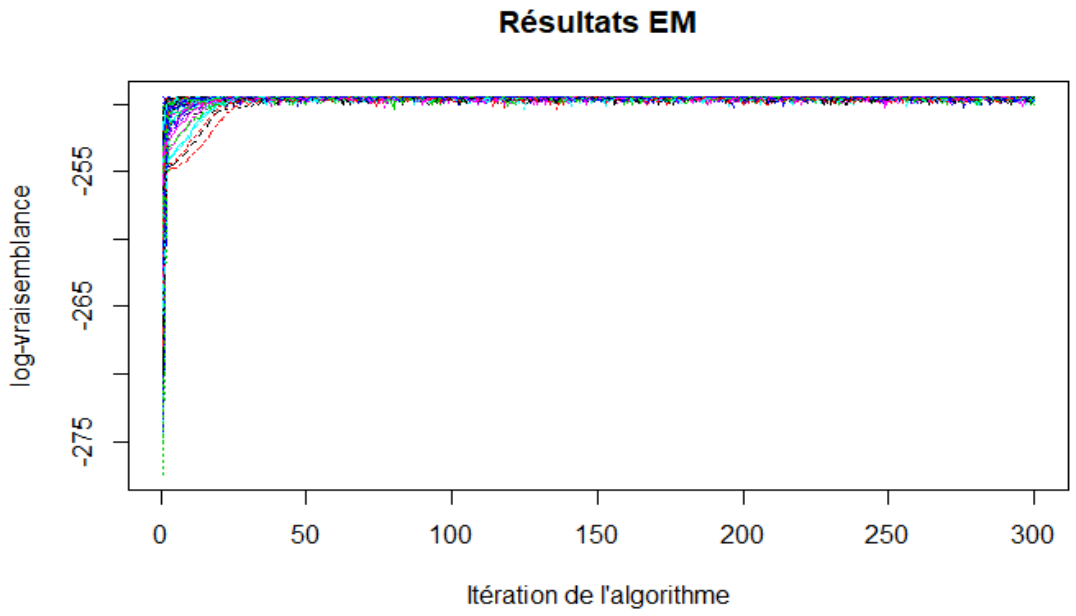


FIGURE 3.5 – Évolution du maximum de la log-vraisemblance de 100 EM.

3.6.2 Vitesse de convergence

Concernant la vitesse de convergence, celle-ci dépend surtout de la taille de l'échantillon et de l'initialisation des paramètres. Plus la taille de l'échantillon est grande, plus l'algorithme semble converger rapidement vers le maximum de la log-vraisemblance. Au contraire, le nombre de modalités p et q fait peu varier la vitesse de convergence. La figure 3.6 montre les résultats de la log-vraisemblance jointe en fonction de la taille de l'échantillon n. Chaque courbe de la figure 3.6 a été réalisée sur un modèle où $p = 2$ et $q = 3$. Pour chacun des modèles simulés, seule la taille de l'échantillon varie avec $n \in \{15, 30, 50, 70, 100, 200, 500, 1000, 2000\}$. On constate alors que pour de petites tailles

d'échantillons $n \in \{15, 30\}$, l'algorithme converge un peu plus lentement que pour de grande taille d'échantillon. On notera qu'à partir de $n = 70$, l'échantillon converge très rapidement vers le maximum de la log-vraisemblance. La vitesse de convergence a

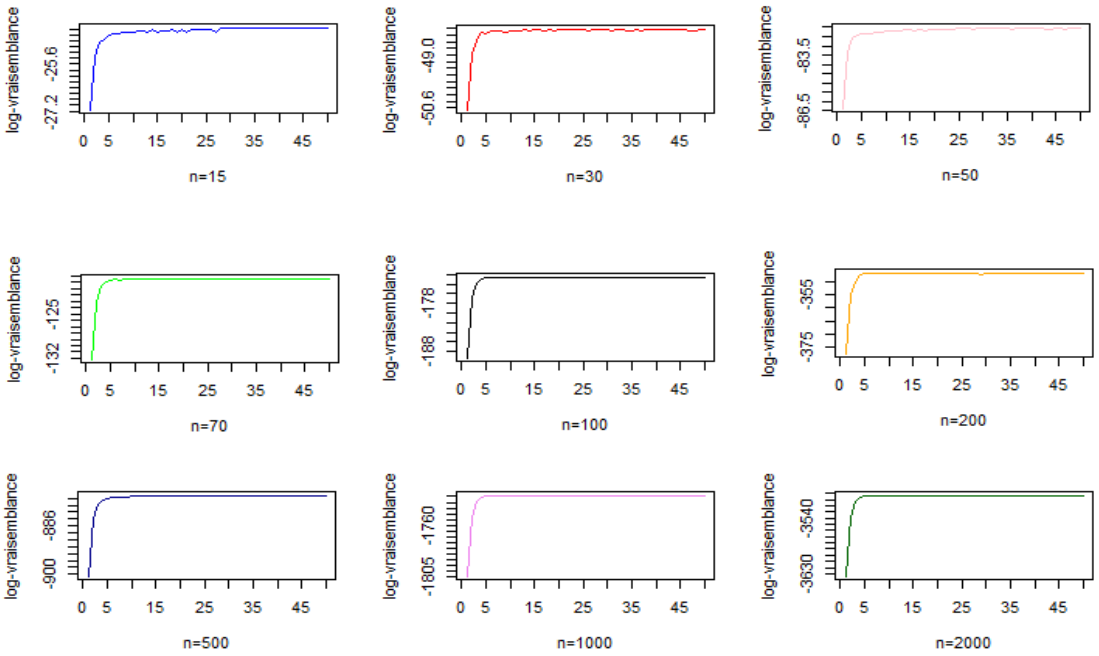


FIGURE 3.6 – Évolution de la convergence en fonction de n

également été étudiée en fonction du nombre de modalités p et q. Pour un jeu de donnée simulé de taille $n = 200$, où les paramètres p et q ont été fixés à différentes valeurs : $(p = 3, q = 4)$, $(p = 3, q = 5)$, où encore $(p = 4, q = 7)$, la vitesse de convergence est similaire.

Convergence des estimateurs pour l'estimation jointe

La figure 3.7, illustre la convergence de l'estimateur p_{13}. Dans la partie 3.4.4 il a été montré qu'empiriquement l'algorithme ne comporte pas de maxima locaux. Pour ces expérimentations, l'algorithme a alors été lancé 5 fois pour chaque taille d'échantillon n. Pour chaque échantillon la meilleure estimation a été sauvegardée et est celle représentée sur la figure 3.7. Chaque courbe de la figure 3.6 a été réalisée sur selon un modèle simulé où $p = 2$ et $q = 3$ avec $\mathbf{p} = (0.6, 0.4)$, $\mathbf{p}_1 = (0.7, 0, 0.3)$ et $\mathbf{p}_2 = (0, 0.8, 0.2)$. Pour chaque

échantillon, seule la taille varie avec $n \in \{150, 200, 500, 1000, 2000, 5000, 10000, 50000\}$. La figure 3.7 montre la convergence de l'estimateur $\hat{p}_{13}$, où la valeur simulée est $p_{13} = 0.3$ pour chaque d'échantillon. Il est alors possible de constater que plus le taille de l'échantillon n est grande, plus l'estimateur $\hat{p}_{13}$ converge vers la valeur simulée avec $\hat{p}_{13} \approx 0.3$, notamment à partir $n = 500$.

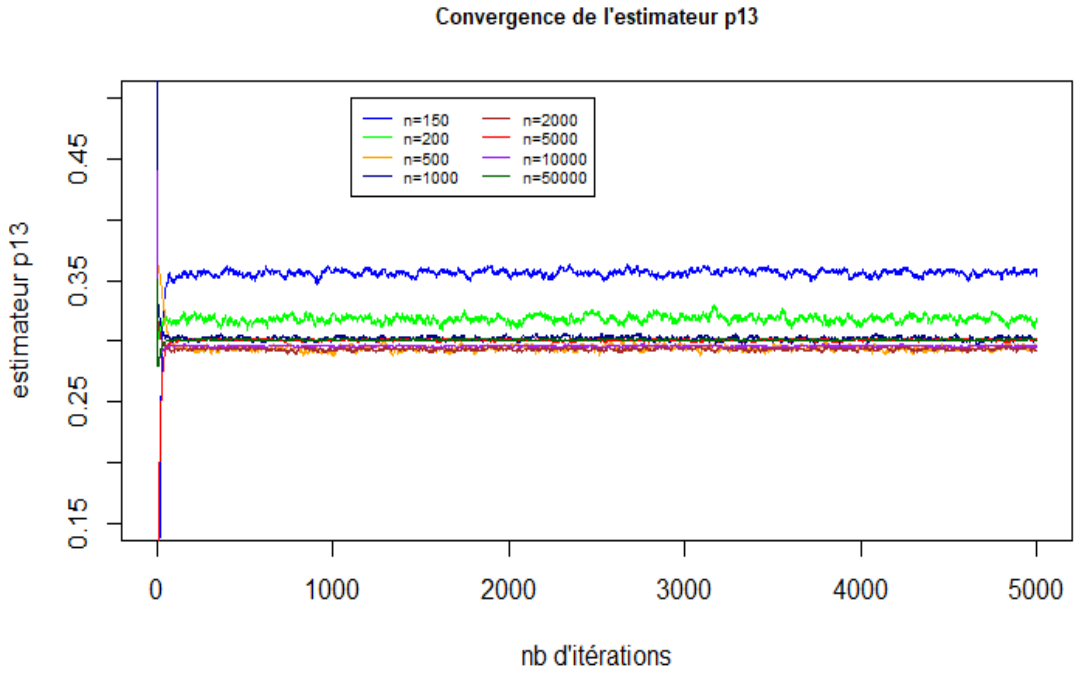


FIGURE 3.7 – Évolution de la convergence de l'estimateur p_{13} en fonction de n

3.7 Conclusion

Dans ce chapitre nous proposons une méthode pour répondre à la problématique énoncée dans le chapitre 1. La principale problématique repose sur le faible nombre d'observations disponible après la modification des descripteurs d'une variable catégorielle. Pour pallier à ce problème, nous proposons de réaliser un transfert de connaissance entre les données avant et après la modification des descripteurs de la variable. L'objectif principal est alors d'utiliser la variable $\boldsymbol{x}$ afin de générer des échantillons $\hat{\boldsymbol{y}}$ de taille suffisante pour permettre la réalisation des différentes analyses statistique telle que de la classification. Les données manquantes inhérentes au contexte de ce travail implique

une grande part d'incertitude. Nous proposons alors de nous placer dans un cadre probabiliste pour pouvoir quantifier cette incertitude. La loi jointe des données permettant de générer l'ensemble des liens stochastiques entre les variables, nous proposons de l'utiliser pour notre modélisation. D'autre part, la loi jointe des données a pour intérêt de faire apparaître la loi marginale $P(\boldsymbol{x})$ dont nous voulons utiliser la connaissance et la loi de probabilités conditionnelles $P(\boldsymbol{y}|\boldsymbol{x})$, inconnue. La suite de ce chapitre concerne l'estimation des paramètres de cette loi. Les données manquantes du problème implique que la jointe des données est inconnue et que le modèle initial proposé soit non identifiable. Afin de le rendre identifiable, des contraintes simples, de type binaires ont été proposées. Ces contraintes consistent à fixer certaines probabilités de transition à zéro et ont l'avantage d'être aisément interprétables dans le modèle final. Deux méthodes sont ensuite proposées pour l'estimation des paramètres. Une estimation par vraisemblance profilée est proposée dans un premier temps afin de "concentrer" la fonction de log-vraisemblance sur les paramètres $\mathbf{p}_{\cdot}$ qui nous intéressent particulièrement. Néanmoins cette méthode suppose que les paramètres $\mathbf{p}$ ne dépendent que des données $\mathbf{x}^{-}$, or le modèle $P(\boldsymbol{y}, \mathbf{p}, \mathbf{p}_{\cdot})$ indique les paramètres $\mathbf{p}$ dépendent également des données $\mathbf{y}^{+}$. Une seconde méthode, permettant l'estimation jointe des paramètres $\mathbf{p}$, $\mathbf{p}_{\cdot}$ a alors été proposée. Les performance des deux méthodes d'estimation ont ensuite été comparées à travers différents jeux de données simulés. La comparaison des deux méthodes montrent que celles-ci retournent des résultats similaires. Cela s'explique par la faible quantité d'observations $\mathbf{y}^{+}$ qui finalement influence peu l'estimation des paramètres $\mathbf{p}$. Pour la suite de l'étude, la méthode d'estimation jointe est utilisée. Les deux méthodes utilisant les données manquantes $(\mathbf{x}^{+}, \mathbf{y}^{-})$ nécessitent l'utilisation d'un algorithme EM, qui est l'algorithme usuel et connu pour être efficace sur les problèmes comportant des données manquantes. Ce chapitre se termine par des expériences numériques indiquant les performance de notre algorithme dans le cadre de l'estimation jointe des paramètres $\boldsymbol{\theta}$. Celles ci sont réalisées sur des données simulées et montre que l'algorithme EM fourni une estimation précis et très proche des paramètres simulés. D'autre part, avec une taille d'échantillon assez grande ($n \geq 100$), l'algorithme converge rapidement. On notera également que la précision de l'estimation dépend de la taille n de l'échantillon. Cette méthode nous permet d'avoir un ensemble de modèles dont les paramètres sont interprétables facilement. Cependant, la modélisation sous contraintes implique que nous avons désormais un ensemble de modèles noté $\boldsymbol{\Delta}$, composé de paramètres fixés à zéros et de paramètres à estimer. L'algorithme EM, utilisé dans le cadre de l'estimation jointe, permet d'obtenir une bonne estimation des paramètres d'un modèle. Cependant, il est nécessaire d'appliquer cet algorithme sur chacun des modèles l'ensemble. Ce processus

consiste alors en une méthode exhaustive, nommée par la suite **EXsearch**. Le manque d'informations disponibles implique que le modèle correspondant le mieux aux données disponibles parmi cet ensemble est inconnu. L'utilisation d'une méthode de sélection de modèle est alors nécessaire. L'objet du chapitre suivant est de présenter une méthode de sélection de modèle efficace et pertinente selon les différents jeux de données possibles.

Chapitre 4

Sélection de modèles de transfert asymptotique et non asymptotique

> Tous les modèles sont faux, mais certains sont utiles.
>
> *George Box*

Dans le chapitre 3, nous avons proposé de modéliser notre problème par la loi jointe $P(\boldsymbol{x},\boldsymbol{y})$, afin de réaliser un transfert de connaissances de la variable $\boldsymbol{x}$ vers la variable $\boldsymbol{y}$. Cette modélisation permet de répondre aux différents objectifs énoncés dans le chapitre 1. Néanmoins, pour rendre le modèle de transfert identifiable en paramètre, celui-ci a été contraint. Les contraintes fixées, de type binaire, consistent à imposer certaines probabilités de transition à zéro. Néanmoins, les probabilités de transition devant être fixées à zéro sont inconnues. Cela implique de finalement travailler avec un ensemble de modèles de transfert noté $\boldsymbol{\Delta} = \{\delta\}$, où les probabilités de transition fixées à zéro et celles à estimer varient, tel que sur l'exemple représenté par la figure 3.3. L'objectif de ce chapitre est alors de trouver le modèle δ le plus en adéquation avec les données observées de l'ensemble $\boldsymbol{\Delta}$. Dans un premier temps, nous nous intéressons à l'identifiabilité des modèles. Puis, après une présentation des deux critères de choix de modèles asymptotiques usuels (AIC et BIC), le critère asymptotique sélectionné est appliqué sur l'ensemble de modèles $\boldsymbol{\Delta}$. L'utilisation d'un critère asymptotique montrant rapidement ses limites, un critère non asymptotique **BIL** est également proposé. Une dernière partie de ce chapitre concerne la comparaison des performances des deux critères.

4.1 Modèles de transfert non identifiables

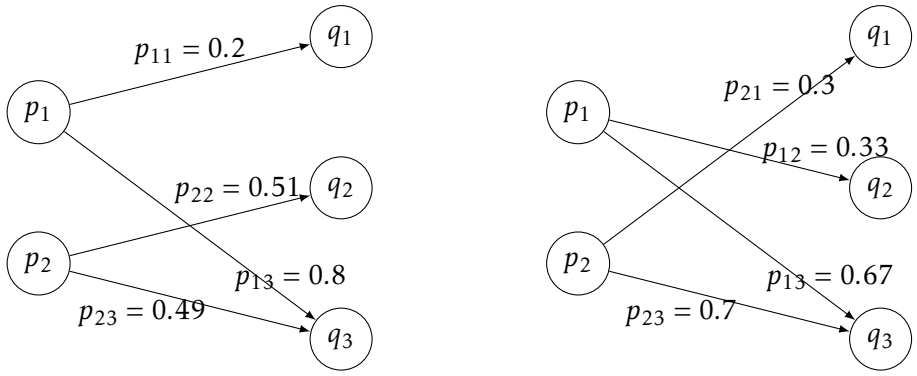
Dans la section 3.3 différentes contraintes ont été proposées afin d'avoir des modèles identifiables en paramètres. La quantité d'information disponible ne permettant pas l'estimation de l'ensemble des probabilités de transition, des contraintes de types binaires ont été proposées. Dans un second temps, des contraintes sur le nombre de probabilités à fixer à zéro ont également été proposées afin de garantir l'identifiabilité en paramètres du modèle. Néanmoins, la faible quantité d'informations disponibles et les contraintes proposées sur les modèles impliquent l'existence de plusieurs modèles possibles, identifiables en paramètres, donnant la même loi $P(\boldsymbol{y})$. Ces modèles ne sont alors pas mathématiquement identifiables dans l'obtention de la loi $P(\boldsymbol{y})$.

4.1.1 Exemple de modèles non identifiables

La figure 4.1 représente deux exemples de modèles, $\boldsymbol{\delta}_1$ et $\boldsymbol{\delta}_2$. Pour ces deux modèles, $p = 2$ et $q = 3$. L'utilisation de la jointe $P(\boldsymbol{x}, \boldsymbol{y})$ implique que ces deux modèles observent la même loi $P(\boldsymbol{x})$, avec $\mathbf{p} = (0.6, 0.4)$. Pour ces deux modèles, seules les probabilités de transition estimées sont représentées par des arcs orientés sur la figure 4.1. Il est alors possible de constater que les probabilités de transition fixées à zéro sont différentes et donc que les deux modèles sont différents. De plus, ces deux modèles respectent les contraintes proposées dans la section 3.3 et sont identifiables en paramètres, avec des probabilités de transition différentes. Or, le calcul de ces deux modèles, faisant partie du même ensemble, amène à la même loi $P(\boldsymbol{y})$, donnée par l'équation 3.4, avec $\mathbf{q} = (0.12, 0.2, 0.68)$, suivant les probabilités de transition indiquées sur les graphes (a) et (b) de la figure 4.1.

4.1.2 Des avantages à la non identifiabilité des modèles

Le fait d'avoir des modèles non-identifiables dans notre ensemble de modèles comporte cependant différents atouts. D'un point de vue technique, le fait que plusieurs modèles soient non identifiables peut permettre de réduire le temps d'exécution de la méthode **EXsearch** proposée. Le nombre de modèles de l'ensemble $\boldsymbol{\Delta}$ pouvant vite devenir conséquent, les étapes d'estimation et de comparaison peuvent vite devenir coûteuses en temps. Or, si certains modèles sont non identifiables, il n'est pas nécessaire de tous les comparer puisqu'ils donnent la même loi $P(\boldsymbol{y})$. D'autre part, le fait que plusieurs modèles donnent la même loi $P(\boldsymbol{y})$ permet de laisser aux métiers la décision finale. Mathématiquement, ces modèles sont identiques. Cependant, d'un point de vue



(a) δ_1 (b) δ_2

FIGURE 4.1 – Exemple de modèles (δ_1 et δ_2) non identifiables.

métier, un de ces modèles est peut être plus intuitif. Cela laisse alors une certaine liberté pour le praticien. La recherche des modèles peut se faire de manière exhaustive ou non exhaustive. Un autre avantage au fait d'avoir des modèles non identifiables concerne la recherche non-exhaustive. En effet, lors d'une recherche non-exhaustive il est possible qu'un seul des modèles non identifiables soit retourné. Or, ce modèle ne sera peut être pas le plus pertinent d'un point de vue métier. Le fait de détecter que d'autres modèles amènent à la même loi $P(\boldsymbol{y})$ peut permettre d'orienter la recherche vers la détection de ces autres modèles afin de laisser le choix aux métiers du modèle final à utiliser. Avoir certains modèles non-identifiables parmi l'ensemble de modèles $\boldsymbol{\Delta}$, a donc des avantages. Il devient alors nécessaire de pouvoir les détecter au sein de l'ensemble de modèles $\boldsymbol{\Delta}$.

4.1.3 Proposition d'une méthodologie pour détecter les modèles non identifiables

De même que pour la partie 3.2.3, dans cette partie nous nous intéressons au modèle de transition définie par l'équation 3.4. Les observations $\mathbf{x}^-$ étant supposées disponibles en quantité suffisante pour bien estimer $\mathbf{p}$, les paramètres $\mathbf{p}$ sont supposés connus et identiques dans chacun des modèles. Deux modèles de transition, tel que définis par l'équation 3.4 seront dit "non identifiable" s'ils respectent la définition suivante :

Définition : Avec $\mathbf{p}$ connus, deux modèles $P(\boldsymbol{y};\mathbf{p}_{.,}\mathbf{p},\delta)$ et $P(\boldsymbol{y};\mathbf{p}'_{.},\mathbf{p},\delta')$ sont dit **non identifiables** en $\mathbf{p}_{.}$ si il existe un couple de vecteurs de paramètres $(\mathbf{p}_{.,}\mathbf{p}'_{.})$ tel que pour deux modèles (δ,δ'), les solutions soient identiques. Autrement dit : $\{(\delta,\delta') \in \Delta^2,\ \exists(\mathbf{p}_{.,}\mathbf{p}'_{.}) \in \mathcal{P}$ tel que $P(.;\mathbf{p}_{.,}\mathbf{p},\delta) = P(.;\mathbf{p}'_{.},\mathbf{p},\delta')$ avec $\delta \neq \delta'$et $\mathbf{p}_{.} \neq \mathbf{p}'_{.}\}$.

A la différence de l'identifiabilité en paramètre des modèles définis en partie 3.2.3, cette définition impose que les deux modèles comparés soient différents. En effet, pour deux modèles identiques, cette définition reviendra à la définition de l'identifiabilité en paramètre. L'objectif est alors de résoudre le système suivant en $p_{hh'}$ et $p'_{hh'}$:

$$\sum_{h=1}^{p} \delta_{hh'} p_{hh'} p_h = \sum_{h=1}^{p} \delta'_{hh'} p'_{hh'} p_h, \quad \forall h' = 1,\dots,q \tag{4.1}$$

sous les contraintes :

$$s.c \begin{cases} \delta_{hh'} = 1 \implies 0 < p_{hh'} <= 1 & \forall h = 1,\dots,p \ \forall h' = 1,\dots,q, \\ \delta'_{hh'} = 1 \implies 0 < p'_{hh'} <= 1 & \forall h = 1,\dots,p \ \forall h' = 1,\dots,q, \\ \delta_{hh'} = 0 \implies p_{hh'} = 0 & \forall h = 1,\dots,p \ \forall h' = 1,\dots,q, \\ \delta'_{hh'} = 0 \implies p'_{hh'} = 0 & \forall h = 1,\dots,p \ \forall h' = 1,\dots,q. \end{cases} \tag{4.2}$$

Les deux modèles mis en comparaison étant connus, les paramètres $\delta_{hh'}$ sont également connus pour les deux modèles. Les deux modèles comparés devant faire partie de l'ensemble de modèles $\mathbf{\Delta}$, ils répondent aux différentes contraintes définies dans la section 3.3. De plus, les paramètres pouvant être estimés et les paramètres étant fixés à zéro sont définis par le modèle et les paramètres $\delta_{hh'}$. En effet, les paramètres fixés à zéro sont définis par $\delta_{hh'} = 0 \implies p_{hh'} = 0$, et les paramètres à estimer par $\delta_{hh'} = 1$. Afin de ne pas modifier les modèles comparés, les paramètres à estimer sont alors nécessairement positifs. D'autre part, travaillant avec des probabilités, les paramètres sont nécessairement inférieurs ou égaux à 1. Ce qui implique que les probabilités de transition différentes de zéro soient comprises dans l'intervalle]0,1]. La détection de modèles non identifiables se fait donc en deux étapes, détaillées dans l'annexe B.

Etape 1 : Utilisation de la forme matricielle L'équation 4.1 correspond à un système linéaire, celui-ci peut être mis sous forme matricielle tel que $\mathbf{A}_{\delta p}\mathbf{P}_{.} = \mathbf{q}_{\delta p}$, où $\mathbf{P}_{.}$ est le vecteur d'inconnues. Les matrices de ce système sont détaillées en annexe B. On notera néanmoins que cette étape concerne uniquement l'équation 4.1, indépendemment des contraintes 4.2. Afin d'avoir une première information sur la solution du système, les propriétés de rang de matrice vont être utilisées. Les propriétés sur les rangs de matrices permettent d'obtenir une condition nécessaire à la non identfiabilité des modèle par l'obtention d'une première information sur la solution du système. Selon les théorèmes de Rouché-Fontené, le calcul du rang de la matrice $\mathbf{A}_{\delta p}$ permet d'avoir une première information sur la solution du système selon trois cas possibles :

Pas de solution : Lorsque que le système n'a pas de solution, la réponse est immédiate : les deux modèles sont identifiables.

Une solution unique : Lorsque la solution est unique, il est nécessaire d'effectuer l'étape 2, vérifiant les contraintes. Si les contraintes sont vérifiées, les deux modèles ne sont pas identifiables.

Infinité de solutions : De même que pour la solution unique, s'il existe une infinité de solutions, l'étape 2 de vérification des contraintes doit être effectuée. A l'instar de la solution unique, si l'une des solutions respecte les contraintes, les deux modèles ne sont pas identifiables.

Etape 2 : Vérification des contraintes Les contraintes 4.2 n'étant pas impliquées dans l'équation 4.1 et la résolution matricielle, lorsqu'une solution existe, il est nécessaire de vérifier que celles-ci soient bien respectées. Une condition suffisante à la non identifiabilité de deux modèles est qu'une des solutions du système donné par l'équation 4.1 respecte les contraintes 4.2. La seconde étape consiste à calculer l'une des solutions du système lorsque la première étape indique qu'il existe une ou plusieurs solutions.

Exemples : Les exemples suivants montrent la nécessité de vérifier le respect des contraintes 4.2.

Exemple 1 : Dans cet exemple, les modèles comparés sont de la forme $p = 2$ et $q = 3$. Les deux modèles de transition comparés sont représentés par les matrices (a) et (b) de la figure 4.2. Les paramètres $\mathbf{p}$ étant connus, seuls les paramètres $\mathbf{p}_.$ sont présentés. Dans ces matrices, les "." signifient que les probabilités de transition sont fixées à zéro.

p_{11}	.	p_{13}
.	p_{22}	p_{23}

(a) δ_1

.	p_{12}	p_{13}
p_{21}	.	p_{23}

(b) δ_2

FIGURE 4.2 – Matrice des modèles de transition δ_1 et δ_2.

L'étape 1 indique que le système correspondant à l'équation (4.1) pour ces modèles possède une infinité de solutions. Il est alors nécessaire d'effectuer l'étape 2 de vérification des contraintes. En posant $\mathbf{p} = (0.6, 0.4)$, une solution obtenue est :
$\mathbf{p}_1 = (0.666 - 0.666r_1, 0, 0.666r_1 + 0.333)$, $\mathbf{p}_2 = (0, 0.51, 0.49)$,
$\mathbf{p}'_1 = (0, 0.333, 0.666)$, $\mathbf{p}'_2 = (0.999 - 0.999r_1, 0, r_1)$,

où r_1 est un paramètre pouvant prendre différentes valeurs, impliquant l'infinité de solutions. Le calcul de cette solution permet de voir que les contraintes sur les paramètres $\mathbf{p}_{.}$ et $\mathbf{p}'_{.}$ sont respectées lorsque le paramètre r_1 appartient à l'intervalle]0,1[. Ce qui implique qu'il existe au moins une solution au système défini par l'équation (4.1) respectant les contraintes (4.2). Les modèles δ_1 et δ_2 sont donc **non identifiables**.

Exemple 2 : Les modèles de transition mis en compétition dans cet exemple sont les modèles représentés par la figure 4.3, δ_1 et δ_3.

p_{11}	.	p_{13}
.	p_{22}	p_{23}

(a) δ_1

p_{11}	p_{12}	.
.	.	p_{23}

(b) δ_3

FIGURE 4.3 – Matrice des modèles de transition δ_1 et δ_3.

L'utilisation des rangs pour résoudre le système correspondant à ces modèles indique qu'il existe une solution unique. A l'instar de l'exemple précédent, il est nécessaire de vérifier que les probabilités de transition respectent les contraintes (4.2). Posant de nouveau $\mathbf{p} = (0.6, 0.4)$, la solution obtenue est la suivante :
$\mathbf{p}_1 = (1.33, 0, -0.33)$, $\mathbf{p}_2 = (0, -0.5, 1.5)$,
$\mathbf{p}'_1 = (1.33, -0.333, 0)$, $\mathbf{p}'_2 = (0, 0, 1)$.
Il peut alors être constaté que l'unique solution du système ne respecte pas les contraintes (4.2). En effet, bien que la solution respecte les modèles définis, aucune des probabilités à estimer n'est comprise dans l'intervalle [0,1].

L'unique solution correspondant au système (4.1) pour les modèles δ_1 et δ_3 ne respectant pas les contraintes 4.2, ces deux modèles sont identifiables.

D'autres exemples sont disponibles dans l'annexe B.

Les modèles se trouvant dans l'ensemble $\mathbf{\Delta}$, ne sont donc pas tous identifiables. Dans cette section, nous proposons une méthodologie pour détecter si deux modèles dans l'ensemble $\mathbf{\Delta}$ sont identifiables ou non. Un algorithme itératif utilisant cette méthode permet par la suite de détecter l'ensemble des modèles non identifiables de $\mathbf{\Delta}$, ou, afin de réduire le temps de calcul, de détecter uniquement les modèles non-identifiables pour en fonction d'un modèle précis. La section suivante se focalise sur la sélection du modèle le plus en adéquation avec nos données d'origines.

4.2 Méthode de sélection de modèles

L'objectif initial de la sélection de modèle est de sélectionner le meilleur modèle parmi un ensemble de modèles. Cependant, la définition de "meilleur" modèle, et les méthodes qui s'y rapportent, varient selon l'objectif final souhaité du modèle. Trois objectifs sont à distinguer :

Prédiction : Le meilleur modèle est celui permettant d'effectuer les meilleures prédictions ;

Classification : Le meilleur modèle est celui permettant de faire de la classification ;

Identification : Le meilleur modèle est celui étant le plus en adéquation avec le modèle ayant servi à générer les données.

L'objectif de ce travail est de trouver le modèle correspondant au mieux aux comportements des internautes, soit aux données disponibles. Dans la suite de ce travail, la définition de "meilleur" modèle correspondant à l'objectif "Identification de modèle" sera alors utilisée car cette étape de choix de modèle est préliminaire à d'autres étapes non définies par avance. Ces étapes peuvent être aussi bien de la prédiction ou de la classification. Le point de vue "identification" est donc un positionnement neutre par rapport aux étapes ultérieures encore non définies au moment de la sélection. Il existe différents critères permettant de faire de la sélection de modèle et répondant à cet objectif. Dans ce chapitre, nous présentons les critères asymptotiques usuels AIC et BIC dont l'objectif est de minimiser un critère pénalisé.

4.2.1 Akaike Information Criterion (AIC)

Le critère AIC (Akaike Information Criterion) est l'un des tous premiers critères apparu dans la littérature en 1973 [2]. Le but de ce critère est la sélection du modèle produisant une bonne approximation de la distribution $(\mathbf{x}^-, \mathbf{y}^+)$ au sens de la divergence de Kullback. Pour cela, une approximation asymptotique de la déviance moyenne est calculée telle que pour un modèle δ on ait :

$$2\mathbb{E}_{\mathbf{x}^-,\mathbf{y}^+,\mathbf{x}^{-'},\mathbf{y}^{+'}}[\log P(\mathbf{x}^{-'}, \mathbf{y}^{+'}) - \log P(\mathbf{x}^{-'}, \mathbf{y}^{+'}; \hat{\boldsymbol{\theta}}_{\mathbf{x}^-,\mathbf{y}^+,\delta})], \tag{4.3}$$

où $\mathbf{x}^-,\mathbf{y}^+$ et $\mathbf{x}^{-'},\mathbf{y}^{+'}$ sont deux échantillons indépendants de même taille et $\hat{\boldsymbol{\theta}}_{\mathbf{x}^-,\mathbf{y}^+,\delta}$ l'estimateur du maximum de vraisemblance de $\boldsymbol{\theta}$ calculé pour l'échantillon $\mathbf{x}^-,\mathbf{y}^+$ et le

modèle δ. Venant de cette approximation, le critère AIC est alors défini par :

$$\mathrm{AIC}(\delta) = -2\log(L_\delta(\mathbf{x}^-, \mathbf{y}^+; \hat{\theta}_\delta)) + 2\nu_\delta, \tag{4.4}$$

où ν_δ est le nombre de paramètres à estimer pour le modèle δ. Le modèle retenu sera le modèle minimisant ce critère. Cette approximation requière les mêmes conditions de régularité que celles requises pour l'obtention de la normalité asymptotique de l'EMV. Ce critère permet de sélectionner le modèle le plus efficace. C'est à dire le modèle ayant le meilleur compromis "biais-variance" dans un objectif de prédiction. Cependant, malgré la pénalisation des modèles comportant un trop grand nombre de paramètres, ce critère peut sélectionner avec une probabilité non nulle des modèles trop complexes. Le critère AIC n'est donc pas consistant [68].

4.2.2 Bayesian Information Criterion (BIC)

Le critère BIC [68] (Bayesian Information Criterion) a été introduit par Schwarz en 1978 [100]. Ce critère se place dans le cadre bayésien de la sélection de modèle où les paramètres θ_δ et les différents modèles δ d'une collection finie de modèles $\{\delta_1, \ldots, \delta_m\}$ sont vus comme des variables aléatoires munies des distributions *a-priori* $P(\theta_\delta|\delta)$ et respectivement $P(\delta)$. Avec le critère AIC, c'est l'un des critères les plus utilisés en statistique. A l'instar du critère AIC, le critère BIC utilise le maximum de vraisemblance. Le critère BIC est une approximation du calcul de la vraisemblance intégrée, conditionnellement au modèle fixé, notée $P(\mathbf{x}^-, \mathbf{y}^+|\delta)$. Cette approximation, pour un modèle δ donné, s'obtient par l'intégration de la distribution jointe du vecteur de paramètres θ et des données $\mathbf{x}^-, \mathbf{y}^+$ sur toutes les valeurs des paramètres θ tel que :

$$P(\mathbf{x}^-, \mathbf{y}^+|\delta) = \int_\theta P(\mathbf{x}^-, \mathbf{y}^+, \theta|\delta) d\theta. \tag{4.5}$$

Cette approche a pour avantage de permettre de donner un poids plus important à certains modèles en prenant en compte les informations que peut détenir l'utilisateur. L'objectif du critère BIC revient au final à sélectionner le modèle δ maximisant la probabilité *a posteriori* $P(\delta|\boldsymbol{x}^-, \boldsymbol{y}^+)$ tel que :

$$P(\delta|\mathbf{x}^-, \mathbf{y}^+) \propto P(\mathbf{x}^-, \mathbf{y}^+|\delta)P(\delta). \tag{4.6}$$

Remarque : La distribution $P(\delta)$ posée sur les modèles est généralement non informative, c'est-à-dire $P(\delta_1) = P(\delta_2) = \ldots = P(\delta_m)$. L'équation 4.6 revient alors à sélectionner le

modèle tel que :

$$\hat{\delta}_{BIC} = \underset{\delta}{\operatorname{argmax}} P(\delta|\mathbf{x}^-, \mathbf{y}^+). \tag{4.7}$$

Néanmoins, le calcul exact de la vraisemblance intégrée $P(\mathbf{x}^-, \mathbf{y}^+|\delta)$ étant rarement possible, certains auteurs tel que Raftery (1995) [90] proposent l'utilisation d'une approximation asymptotique. Pour faire cette approximation, la méthode d'approximation de Laplace est utilisée.

Approximation de Laplace Soit une fonction $L : \mathbb{R}^d \to \mathbb{R}$ telle que L soit deux fois différenciable de $\mathbb{R}^d$ et atteint un unique maximum sur $\mathbb{R}^d$ en u^*. Alors

$$\int_{\mathbb{R}^d} \exp^{nL(u)} = \exp^{nL(u^*)} (\frac{2\pi}{n})^{\frac{d}{s}} |-L''(u^*)|^{-\frac{1}{2}} + \mathcal{O}(n^{-1}). \tag{4.8}$$

Pour toutes fonctions dépendant de n, sous certaines conditions, ce résultat reste valable [68]. En posant $L = \frac{1}{n}G$, et $G = \log(P(\mathbf{x}^-, \mathbf{y}^+|\theta_\delta)$, la probabilité $P(\mathbf{x}^-, \mathbf{y}^+|\delta)$ peut alors s'écrire :

$$P(\mathbf{x}^-, \mathbf{y}^+|\delta) = \exp^{G(\theta^*_\delta)} (\frac{2\pi}{n})^{\frac{\nu_\delta}{2}} |A_{\theta^*_\delta}|^{-\frac{1}{2}} + \mathcal{O}(n^{-1}). \tag{4.9}$$

où ν_δ correspond au nombre de paramètres libres du modèle δ et A_θ est l'opposée de la matrice hessienne des dérivées secondes partielles de L. Asymptotiquement, l'approximation (4.10) peut alors être obtenue.

$$\log P(\mathbf{x}^-, \mathbf{y}^+|\delta) \approx \log(L_\delta(\hat{\theta}_\delta; \mathbf{x}^-, \mathbf{y}^+|\delta)) - \frac{\nu_\delta}{2} \log(n). \tag{4.10}$$

Pour obtenir cette approximation un passage au logarithme et les conditions de régularité de l'EMV (section 3.4.1) et de la matrice A sont nécessaires. Dans cette approximation $\hat{\theta}_\delta$ correspond à l'estimateur du maximum de vraisemblance de θ_δ. De l'équation (4.10), le critère BIC pour un modèle δ donné est alors défini par :

$$\mathrm{BIC}_\delta = -2\log(L_\delta(\hat{\theta}_\delta; \mathbf{x}^-, \mathbf{y}^+)) + \nu_\delta \ln(n), \tag{4.11}$$

que l'on cherche à minimiser, tel que le modèle sélectionné soit : $\delta_{BIC} = \mathrm{argmin}_\delta \mathrm{BIC}_\delta$. L'approximation asymptotique du critère BIC cherche donc le modèle minimisant l'équation (4.11). Le terme en $\ln(n)$ de l'équation (4.11) assure que le critère BIC permet de converger vers le quasi-vrai modèle [68], c'est-à-dire le modèle étant le plus en adéquation avec le modèle ayant servi à générer les données, dans la liste de modèles considérée et sachant les données. Le critère BIC a pour but de sélectionner le quasi-vrai modèle d'un ensemble alors que le critère AIC a pour but de sélectionner le modèle

ayant le meilleur compromis "biais-variance". Les objectifs de ces deux critères sont donc bien différents, le critère AIC ayant un objectif de prédiction alors que le critère BIC a un objectif d'identification. D'autre part, le critère BIC est consistant alors le critère AIC ne l'est pas.

La modélisation réalisée dans le chapitre 3 implique de travailler avec un ensemble de modèles, noté $\mathbf{\Delta}$ à comparer. L'objectif de ce travail est de trouver le modèle correspondant le mieux à nos données initiales et contraintes. C'est à dire de retrouver le quasi-vrai modèle de l'ensemble $\mathbf{\Delta}$. L'utilisation du critère BIC pour la sélection de modèle semble alors la plus pertinente pour répondre à notre objectif d'identification. Appliquée à nos modèles, l'équation (4.11) est définie par :

$$\mathrm{BIC}_\delta = -2\ell_\delta(\hat{\boldsymbol{\theta}}; \mathbf{x}^-, \mathbf{y}^+) + \nu_\delta \ln(n) \tag{4.12}$$

où ν_δ correspond au nombre de paramètres libres du modèle δ.

Par la suite, la stratégie utilisant la méthode **EXsearch** et ce critère de sélection de modèle sera appelée **EXBIC**. Cette stratégie a fait l'objet d'une communication lors de la conférence JDS 2016 [8]

4.2.3 Limites du BIC Approché

Petits échantillons

L'approximation asymptotique utilisée dans la construction du critère BIC implique une convergence du critère dépendant de n, soit de la taille de l'échantillon. Lorsque la taille de l'échantillon est trop petite, l'approximation peut montrer ses limites et être mauvaise (Keribin (2010)[63]). L'objectif de ce travail est de trouver un modèle pertinent afin d'apparier les échantillons avant et après modification des descripteurs dans le but de garder des analyses pertinentes et robustes. Cela implique de travailler rapidement après la modification des descripteurs de la variable et donc de travailler potentiellement avec de petits échantillons. Cette limite du critère BIC peut alors rapidement devenir une contrainte à l'utilisation de ce critère.

Paramètres au bord de l'espace

L'approximation de Laplace reposant sur des développements de Taylor [62], les garanties de l'approximation utilisée ne fonctionnent que lorsque les paramètres sont éloignés des bords de l'espace des paramètres. Autrement dit, lorsqu'un paramètre est proche du bord de son espace, aucune garantie théorique n'indique que l'approximation

utilisée par le critère BIC soit valable et que le critère soit utilisable dans ce cas. Les contraintes imposées pour rendre les modèles identifiables en paramètres pour la modélisation proposée sont très fortes. Ces contraintes imposant d'avoir des probabilités de transition fixées à zéro, l'estimation d'un paramètre sur le bord de son espace [0,1] est relativement courante. Le tableau 4.1 montre un exemple de modèle estimé par l'algorithme EM sur un jeu de données réelles, où les "." indiquent les probabilités à zéro.

$X \backslash Y$	T	T+	T++	I	TR	TR+	TR++
T	0.64	.	0.36	.	.	.	.
T++	.	0.90	.	.	.	.	0.10
I	.	.	.	0.74	.	0.26	.
TR	.	.	.	.	1.00	.	.

Tableau 4.1 – Probabilités d'appariement estimées selon le modèle ayant le plus petit BIC.

Il est alors possible de constater qu'une des probabilités de transition est estimée à 1, indiquée en rouge. Il est à noter que l'estimation de ce paramètre à 1 n'est pas imposée par le modèle. En effet, a l'instar des autres modalités de la variable $\boldsymbol{x}$, le modèle initial n'imposait pas de zéro sur l'ensemble de autres modalités de la variable $\boldsymbol{y}$ liées à la modalité (TR) de la variable $\boldsymbol{x}$ pour contraindre l'estimation de ce paramètre à 1. L'estimation de ce paramètre à 1 est le résultat de l'algorithme EM. Dans l'espace de modèle $\boldsymbol{\Delta}$ plusieurs modèles ont cette particularité, dûe aux contraintes fixées. Sur ce jeu de données, l'utilisation du critère BIC n'est peut-être pas la plus pertinente, la sélection du quasi-vrai modèle par le critère n'étant plus théoriquement garantie. D'autre part, il est courant que seulement une partie des modalités de la variable soit modifiée. Le modèle est alors semi-déterministe et certaines probabilités de transition du modèle sont alors estimées sur le bord de l'espace des paramètres. Un second exemple est donné par le tableau 4.2. Ce tableau reprend l'exemple présenté en section 1.1.4 par la figure 1.4 pour la modification datant du 07 novembre 2017. Dans cet exemple, une seule modalité a été supprimée. Les autres modalités étant identiques, il parait naturel que les probabilités de transition associées, non modifiées, soient estimées à 1. Ce qui signifie que les probabilités sont estimées sur le bord de l'espace des paramètres. De nouveau, l'utilisation du critère BIC n'est peut-être pas la plus pertinente, la sélection du quasi-vrai modèle par le critère n'étant plus théoriquement garantie. A l'instar de l'approximation asymptotique, l'approximation de Laplace peut vite devenir une contrainte à l'utilisation du critère BIC dans notre cadre où les limites de ce critère peuvent rapidement être atteintes.

Les différentes limites liées à la construction du critère BIC et ses approximations

$X \backslash Y$	T	T++	I	TR	TR+	TR++
T	1.00	.	.	.	.	.
T+	0.4	0.6	.	.	.	.
T++	.	0.8	0.2	.	.	.
I	.	.	1.00	.	.	.
TR	.	.	.	1.00	.	.
TR+	.	.	.	.	1.00	.
TR++	.	.	.	.	.	1.00

Tableau 4.2 – Probabilités d'appariements estimées selon le modèle ayant le plus petit BIC.

sont négligeables pour de nombreuses applications. Cependant, le cadre de ce travail et la modélisation réalisée dans la section 3.2, notamment les contraintes imposées sur les modèles, ses limites se révèlent trop souvent atteintes. Pour pallier à ces limitations, liées aux approximations réalisées dans la construction du critère, nous proposons un nouveau critère. Ce critère, notée **BIL** (Bayesian Intégrated Likelihood) repose sur le calcul de la vraisemblance intégrée de manière exacte. La section suivante présente le critère **BIL**.

4.3 Bayesian Integrated Likelihood (BIL)

Le premier critère choisi pour réaliser la sélection de modèle était le critère asymptotique BIC. Cependant, les limites liées aux approximations dans la construction de ce critère deviennent rapidement trop contraignantes lorsque ce critère est appliqué à notre problème. Afin de pallier à ces contraintes, nous proposons de calculer un nouveau critère, le critère **BIL**, reposant sur le calcul de la vraisemblance intégrée $P(\mathbf{x}^-, \mathbf{y}^+|\delta)$ des modèles δ de manière exacte. De même que pour le critère BIC asymptotique, nous nous plaçons donc dans la cadre bayésien. Les paramètres θ et les différents modèles δ sont donc supposés être des variables aléatoires munies de distribution a priori. L'intérêt de l'inférence bayésienne est de pouvoir prendre en compte une information a priori, dans des lois $P(\theta|\delta)$ et $P(\delta)$, provenant d'expériences passées afin d'étudier la loi a postériori de δ connaissant les observations de $\mathbf{x}^-, \mathbf{y}^+$. A l'instar du critère BIC défini dans la section 4.2.2, la loi a priori sur les modèles $P(\delta)$ est supposée non informative. La vraisemblance intégrée des modèles $P(\mathbf{x}^-, \mathbf{y}^+|\delta)$ étant désormais calculée de manière non asymptotique, il est nécessaire de définir précisément les lois a priori des paramètres $P(\theta|\delta)$ utilisées. Une fois les lois a priori définies, le calcul de la vraisemblance intégrée peut être réalisé par approximation non asymptotique à travers deux étapes : L'intégration exacte de la

distribution des données complètes sur les paramètres, suivie par une approximation de la somme sur toutes les valeurs possibles pouvant être prises par les individus $\mathbf{x}^+$ ($\mathbf{x}^+ \in \mathcal{X}$). Le but est alors d'obtenir la distribution marginale des données observées. Pour effectuer cette approximation, une stratégie Bayésienne d'échantillonnage préférentiel est utilisée. La distribution instrumentale Bayésienne qui lui est associée, est dérivée de manière naturelle en utilisant le fait que l'inférence bayésienne est efficacement implémentée à travers un échantillonneur de Gibbs grâce aux propriétés de conjugaison car nous allons nous appuyer sur des lois a priori conjuguées. La section suivante présente les lois a priori utilisées pour les paramètres $P(\boldsymbol{\theta}|\boldsymbol{\delta})$.

4.3.1 Loi a priori

Avant toute inférence Bayésienne, il est nécessaire de définir précisément les distributions a priori sur les paramètres, avec $\boldsymbol{\theta} = (\mathbf{p}, \mathbf{p}_.)$. Dans un premier temps, l'indépendance classique entre chaque vecteur de paramètres est supposée, ce qui mène à :

$$P(\boldsymbol{\theta}) = P(\mathbf{p}) \prod_{h=1}^{p} P(\mathbf{p}_h). \tag{4.13}$$

Afin de faciliter le calcul de la loi a posteriori, donnée par l'équation (4.6), un choix naturel de lois a priori est celui des lois conjuguées [93], dont la définition est la suivante :

Définition : [93] Une famille $\mathcal{F}$ de distributions sur Θ est dite conjuguée par une fonction de vraisemblance $f(x|\theta)$ si pour tout $\pi \in \mathcal{F}$, la distribution a posteriori $\pi(.|x)$ appartient également à $\mathcal{F}$.

Les lois $P(\mathbf{x}^-)$ et $P(\mathbf{y}^+|\mathbf{x}^-)$ étant des lois multinomiales, la loi conjuguée naturelle correspondante est une loi de Dirichlet. L'a priori devant intervenir de façon minimale dans la loi a posteriori, le classique Prior conjuguée non informatif de Jeffreys ([59] Jeffrey 1946) est alors utilisé, pour les paramètres $\mathbf{p}$. Cette loi a la particularité de pondérer les valeurs extrêmes. La loi a priori des paramètres $\mathbf{p}$ est alors définie par :

$$P(\mathbf{p}) = \mathcal{D}_p\left(\frac{1}{2}, \ldots, \frac{1}{2}\right). \tag{4.14}$$

Les paramètres $\mathbf{p}$ ne dépendant pas du modèle $\boldsymbol{\delta}$, ce prior est indépendant du modèle $\boldsymbol{\delta}$. A l'inverse, les paramètres $\mathbf{p}_h$ étant dépendant du modèle $\boldsymbol{\delta}$, le prior $P(\mathbf{p}_h)$ en est dépendant. De nouveau, un prior de Dirichlet conjugué sera utilisé, mais celui-ci sera dégénéré lorsque le paramètre $\delta_{hh'} = 0$ et correspondra à un prior conjugué non

informatif de Jeffreys sinon. Le prior $P(\mathbf{p}_h)$ est défini par :

$$P(\mathbf{p}_h) = \mathcal{D}_q\left(\frac{1}{2}\delta_{h1}, \ldots, \frac{1}{2}\delta_{hq}\right). \tag{4.15}$$

La sélection de modèle dans un cadre Bayésien repose sur le calcul de la vraisemblance des données observées $P(\mathbf{x}^-, \mathbf{y}^+)$, communément appelée vraisemblance marginale ou intégrée. Les différents priors étant définis, le calcul de la vraisemblance des données observées peut désormais être réalisé, le but étant désormais d'exprimer $P(\mathbf{x}^-, \mathbf{y}^+)$.

4.3.2 Vraisemblance marginale

La règle de Bayes établit que la loi a posteriori peut être définie par :

$$P(\delta|\mathbf{x}^-, \mathbf{y}^+) = \frac{P(\mathbf{x}^-, \mathbf{y}^+|\delta)P(\delta)}{P(\mathbf{x}^-, \mathbf{y}^+)}, \tag{4.16}$$

généralement écrite sous forme de proportionnalité telle que définie par l'équation (4.6) La loi a priori $P(\delta)$ étant supposée non informative (uniforme), l'objectif désormais est de calculer $P(\mathbf{x}^-, \mathbf{y}^+|\delta)$, aussi noté $P(\mathbf{x}^-, \mathbf{y}^+)$ lorsque la notation du modèle δ est implicite.

Intégration approchée de la vraisemblance des données observées

Le calcul de la vraisemblance intégrée des données observées peut se faire par l'obtention de la loi marginale sur $\boldsymbol{\theta}$ dans une forme explicite et le calcul d'une somme sur les données manquantes ($\mathbf{x}^+$,$\mathbf{y}^-$). Le fait que le modèle sur les données complètes ($\mathbf{x}^-$,$\mathbf{x}^+$,$\mathbf{y}^-$,$\mathbf{y}^+$) et que la loi a priori soit conjugués permet l'obtention de la loi marginale sur $\boldsymbol{\theta}$ [93]. Notant par $\mathcal{X}$ toutes les valeurs possibles de $\mathbf{x}^+$ et par $\mathcal{Y}$ toutes les valeurs possibles de $\mathbf{y}^-$, l'équation suivante est obtenue :

$$P(\mathbf{x}^-, \mathbf{y}^+) = \sum_{\mathbf{x}^+ \in \mathcal{X}, \mathbf{y}^- \in \mathcal{Y}} P(\mathbf{x}^-, \mathbf{y}^-, \mathbf{x}^+, \mathbf{y}^+), \tag{4.17}$$

avec

$$P(\mathbf{x}, \mathbf{y}) = \iint_{\mathcal{P}, \mathcal{P}_\cdot} P(\mathbf{x}, \mathbf{y}, \mathbf{p}, \mathbf{p}_\cdot) d\mathbf{p} d\mathbf{p}_\cdot, \tag{4.18}$$

correspondant à la vraisemblance intégrée des données complètes, où $\mathcal{P}$ est l'espace des paramètres $\mathbf{p}$ et $\mathcal{P}_\cdot$ est l'espace des paramètres $\mathbf{p}_\cdot$. On notera que contrairement au cadre fréquentiste, dans le cadre bayésien les paramètres $\mathbf{p}$ et $\mathbf{p}_\cdot$ sont supposés être des variables aléatoires. Utilisant l'hypothèse d'indépendance entre les vecteurs de

paramètres, l'équation 4.17 devient :

$$P(\mathbf{x}^-,\mathbf{y}^+) = \sum_{\mathbf{x}^+\in\mathcal{X},\mathbf{y}^-\in\mathcal{Y}} \iint_{\mathcal{P},\mathcal{P}_.} P(\mathbf{y}^-|\mathbf{x}^-,\mathbf{p}_.)P(\mathbf{x}^-,\mathbf{p})P(\mathbf{y}^+|\mathbf{x}^+,\mathbf{p}_.)P(\mathbf{x}^+,\mathbf{p})d\mathbf{p}d\mathbf{p}_. \quad (4.19)$$

L'équation 4.19 permet de remarquer que le calcul de la somme sur $\mathcal{Y}$ se simplifie. Aucune donnée de cette équation ne dépend des données $\mathbf{y}^-$ qui n'apparaissent que pour le calcul de la probabilité $P(\mathbf{y}^-|\mathbf{x}^-)$. La somme sur $\mathcal{Y}$ pour le calcul de $P(\mathbf{y}^-|\mathbf{x}^-)$ est alors égale à 1. L'équation 4.19 peut donc être simplifiée. Utilisant la vraisemblance intégrée des données complètes $P(\mathbf{x},\mathbf{y}^+)$, où $\mathbf{x} = (\mathbf{x}^-,\mathbf{x}^+)$, il est possible de calculer la vraisemblance intégrée des données observées $P(\mathbf{x}^-,\mathbf{y}^+)$ de manière exacte. Pour cela il est nécessaire de calculer de manière exacte la vraisemblance intégrée des données complètes $P(\mathbf{x},\mathbf{y}^+)$ [13].

Vraisemblance intégrées des données complètes

La vraisemblance intégrée des données complètes est définie par

$$P(\mathbf{x},\mathbf{y}^+) = \int_\Theta P(\mathbf{x}^+,\mathbf{x}^-,\mathbf{y}^+|\boldsymbol{\theta})P(\boldsymbol{\theta})d(\boldsymbol{\theta}), \quad (4.20)$$

où $\boldsymbol{\theta}$ est l'espace de l'ensemble des paramètres. Avec l'hypothèse d'indépendance stochastique, conditionnelle aux paramètres $\boldsymbol{\theta}$, des couples de données $(\mathbf{x}^-,\mathbf{y}^-)$ et $(\mathbf{x}^+,\mathbf{y}^+)$, défini dans la section 3.1.2, il vient :

$$\begin{aligned} P(\mathbf{x},\mathbf{y}^+) &= \iint_{\mathcal{P},\mathcal{P}_.} P(\mathbf{x},\mathbf{y}^+,\mathbf{p},\mathbf{p}_.)d\mathbf{p}d\mathbf{p}_. \\ &= \iint_{\mathcal{P},\mathcal{P}_.} P(\mathbf{x},\mathbf{y}^+|\mathbf{p},\mathbf{p}_.)P(\mathbf{p},\mathbf{p}_.)d\mathbf{p}d\mathbf{p}_. \\ &= \underbrace{\int_{\mathcal{P}_.} P(\mathbf{y}^+|\mathbf{x}^+,\mathbf{p}_.)P(\mathbf{p}_.)d\mathbf{p}_.}_{A} \times \underbrace{\int_{\mathcal{P}} P(\mathbf{x}|\mathbf{p})P(\mathbf{p})d\mathbf{p}}_{B} . \end{aligned}$$

Dans la section 4.3.1, les distributions a priori ont été définies comme étant des distributions non informatives de Jeffrey. Les distributions a priori non informatives conjuguées de Jeffrey étant disponibles pour l'ensemble des paramètres, il est possible de calculer la vraisemblance des données complètes sous une forme explicite. Exprimant A tel que $I_h = \{i : x_{ih}^+ = 1\}$ indique le nombre d'individus dans $\mathbf{x}^+$ ayant le niveau h, on obtient :

$$A = \int_{\mathcal{P}_\cdot} \left\{ \prod_{i=1}^{n^+} P(\boldsymbol{y}_i^+ | \boldsymbol{x}_i^+, \mathbf{p}_\cdot) \right\} P(\mathbf{p}_\cdot) d\mathbf{p}_\cdot \tag{4.21}$$

$$= \int_{\mathcal{P}_\cdot} \left\{ \prod_{h=1}^{p} \prod_{i \in I_h} P(\boldsymbol{y}_i^+ | \boldsymbol{x}_i^+, \mathbf{p}_\cdot) \right\} P(\mathbf{p}_\cdot) d\mathbf{p}_\cdot \tag{4.22}$$

$$= \int_{\mathcal{P}_\cdot} \left\{ \prod_{h=1}^{p} \prod_{i \in I_h} P(\boldsymbol{y}_i^+ | \boldsymbol{x}_i^+, \mathbf{p}_h) \right\} \left\{ \prod_{h=1}^{p} P(\mathbf{p}_h) \right\} d\mathbf{p}_h \tag{4.23}$$

$$= \int_{\mathcal{P}_\cdot} \prod_{h=1}^{p} \left\{ \prod_{i \in I_h} P(\boldsymbol{y}_i^+ | \boldsymbol{x}_i^+, \mathbf{p}_h) P(\mathbf{p}_h) d\mathbf{p}_h \right\} \tag{4.24}$$

$$= \prod_{h=1}^{p} \underbrace{\int_{\mathcal{P}_\cdot} \prod_{i \in I_h} P(\boldsymbol{y}_i^+ | \boldsymbol{x}_i^+, \mathbf{p}_h) P(\mathbf{p}_h) d\mathbf{p}_h}_{A_h} . \tag{4.25}$$

De l'équation 4.25, le terme A_h est alors une simple application de la propriété conjuguée d'un modèle multinomial (voir par exemple Robert [93]). Son expression est alors définie par :

$$\boxed{A_h = \frac{\Gamma(\frac{q_h}{2})}{\Gamma(\frac{1}{2})^2} \times \frac{\prod_{h'=1}^{q} \left[\Gamma(n_{hh'}{}^+ + \frac{1}{2}) \right]^{\delta_{hh'}}}{\Gamma(n_h{}^+ + \frac{q_h}{2})} \times \prod_{i=1}^{n} \prod_{h'=1}^{q} (\delta_{hh'})^{(x_{ih} y_{ih'}^+)}} \tag{4.26}$$

où $q_h = \sum_{h'=1}^{q} \delta_{hh'}$, $n^+ = \#I_h$ et $n_{hh'}^+ = \#\{i \in I_h : \delta_{hh'} = 1\}$. Similairement, le terme B est également une application de la propriété conjuguée d'un modèle multinomial, et peut être exprimé par :

$$\boxed{B = \frac{\Gamma(\frac{p}{2})}{\Gamma(\frac{1}{2})^p} \times \frac{\prod_{h=1}^{p} \Gamma(n_h + \frac{1}{2})}{\Gamma(n + \frac{p}{2})}} . \tag{4.27}$$

La vraisemblance intégrée des données complètes donnée par l'équation 4.20 est donc calculable de manière exacte. Cependant, le calcul de la vraisemblance intégrée des données observées $P(\mathbf{x}^-, \mathbf{y}^+) = \sum_{\mathbf{x}^+ \in \mathcal{X}} P(\mathbf{x}^-, \mathbf{x}^+, \mathbf{y}^+)$, donnée par l'équation (4.17), nécessite, également, le calcul de la somme où $\mathcal{X}$ correspond à toutes les valeurs possibles de $\mathbf{x}^+$. Le calcul de la somme sur $\mathcal{X}$ incluant trop de termes, elle ne peut être calculée de manière exacte. Pour exemple, pour $n = 5$ avec $p = 2$ et $q = 3$, la somme sur les données

$\mathbf{x}^+$ est de 7776, et pour $n = 5$ avec $p = 3$ et $q = 4$ le résultat est de 248832, ce qui n'est plus faisable de manière analytique ou prend énormément de temps. Suivant Biernacki (2010) [13] et Cassella (2000) [21], une stratégie d'importance sampling peut alors être utilisée pour approcher la somme sur les données $\mathbf{x}^+ \in \mathcal{X}$ et pouvoir ainsi calculer la vraisemblance intégrée des données observées $P(\mathbf{x}^-, \mathbf{y}^+)$ donnée par l'équation 4.19.

4.3.3 Approximation de la vraisemblance marginale $P(\mathbf{x}^-, \mathbf{y}^+)$ par échantillonnage préférentiel

L'obtention de la vraisemblance intégrée des données observées $P(\mathbf{x}^-, \mathbf{y}^+) = \sum_{\mathbf{x}^+ \in \mathcal{X}} P(\mathbf{x}^-, \mathbf{x}^+, \mathbf{y}^+)$, donnée par l'équation 4.17 implique le calcul de la somme sur les données $\mathbf{x}^+ \in \mathcal{X}$. Le calcul de cette somme n'étant pas réalisable de manière exacte, dû au trop grand nombre de termes qu'elle inclut, nous proposons de l'approcher par une stratégie d'échantillonnage préférentiel. L'échantillonnage préférentiel est une méthode utilisée dans les méthodes de Monte Carlo. Avant de définir l'échantillonnage préférentiel, il est donc nécessaire de présenter le principe des méthodes de Monte Carlo.

Méthode de Monte Carlo : [95] Les méthodes de Monte Carlo ont pour essence l'utilisation d'expériences répétées pour évaluer une quantité ou résoudre un système déterministe. Ces méthodes peuvent servir pour :

— le calcul d'intégrale,
— la résolution d'équations aux dérivées partielles,
— la résolution de système linéaire,
— la résolution de problèmes d'optimisation (algorithme du recuit simulé).

Dans le cas présent, la méthode de Monte Carlo est utilisée dans le but de calculer une approximation de la quantité $P(\mathbf{x}^-, \mathbf{y}^+)$, afin d'éviter la calcul de la somme sur les données $\mathbf{x}^+$. Les méthodes de Monte Carlo proposent alors d'estimer la quantité $P(\mathbf{y}^+, \mathbf{x}^-)$, équivalente à l'espérance $\mathbb{E}[P(\mathbf{y}^+, \mathbf{x}^-, \mathbf{x}^+)]$, approchée par :

$$\hat{P}(\mathbf{y}^+, \mathbf{x}^-) = \frac{1}{S} \sum_{s=1}^{S} P(\mathbf{x}^+_{(s)}, \mathbf{y}^+, \mathbf{x}^-) \tag{4.28}$$

avec la génération d'un grand nombre d'échantillons S de données i.i.d $(\mathbf{x}^+_{(s)})_{s=1,\dots,S}$, où les échantillons $\mathbf{x}^+_{(s)}$ sont générés selon la loi uniforme sur $\mathcal{X}$. Les méthodes de Monte Carlo ont pour propriétés :

— l'obtention d'un estimateur $\hat{P}(\mathbf{y}^+, \mathbf{x}^-)$ non biaisé,
— une convergence de l'estimation $\hat{P}(\mathbf{y}^+, \mathbf{x}^-) \underset{n \to \infty}{\longrightarrow} P(\mathbf{y}^+, \mathbf{x}^-)$ avec une probabilité de 1,

— une erreur quadratique de $\frac{1}{\sqrt{n}}\sigma$ avec $\sigma^2 = \text{var}(P(\mathbf{x}^-, \mathbf{y}^+, \mathbf{x}^+))$.
Bien que très efficace, la variance de la quantité estimée, ici $P(\mathbf{y}^+, \mathbf{x}^-)$, ralentie la convergence de l'estimateur. Il a alors été proposé différentes méthodes de réduction de variance, dont la stratégie d'échantillonnage préférentiel, afin d'accélérer la convergence de l'estimateur.

L'échantillonnage préférentiel : *Importance sampling* en Anglais ,[76], [95], est une méthode de réduction de la variance utilisée dans les méthodes de Monte Carlo. L'idée principale de l'échantillonnage préférentiel est d'utiliser une fonction alternative (ou fonction auxiliaire), notée $I_{(\mathbf{x}^-, \mathbf{y}^+)}(\mathbf{x}^+)$, permettant de remplacer les tirages selon la loi uniforme sur $\mathcal{X}$, qui n'avantagent aucune région, par des tirages plus ciblés. Ainsi, l'échantillonnage est fait suivant l'importance de la distribution $I_{(\mathbf{x}^-, \mathbf{y}^+)}(\mathbf{x}^+)$, permettant de concentrer les tirages de l'échantillon sur les régions de haute importance, et d'éviter de tirer des échantillons dans des régions non significatives. L'objectif est ainsi de diminuer la variance σ^2 et d'accélérer la convergence de l'estimateur $\hat{P}(\mathbf{y}^+, \mathbf{x}^-)$. Cette méthode permet de garder l'efficacité des méthodes de Monte Carlo tout en réduisant significativement son temps de calcul. En effet, la fonction d'importance permettant de cibler l'échantillonnage, le nombre de simulations S nécessaire à la convergence de l'estimation $\hat{P}(\mathbf{y}^+, \mathbf{x}^-)$ est alors réduit.

La fonction d'importance utilisée pour notre stratégie d'échantillonnage préférentiel, notée $I_{(\mathbf{x}^-, \mathbf{y}^+)}(\mathbf{x}^+)$, correspond à la fonction de densité de probabilité des données $\mathbf{x}^+$. Cette densité, pouvant dépendre des données $\mathbf{x}^-$ et des données $\mathbf{y}^+$, doit avoir les propriétés suivantes :

— $\sum_{\mathbf{x}^+ \in \mathcal{X}} I_{(\mathbf{x}^-, \mathbf{y}^+)}(\mathbf{x}^+) = 1$
— $I_{(\mathbf{x}^-, \mathbf{y}^+)}(\mathbf{x}^+) \geq 0, \ \forall \mathbf{x}^+$.

Remarque : Le support de cette densité inclut nécessairement le support de $P(\mathbf{x}, \mathbf{y}^+)$. Utilisant l'approximation de Monte Carlo [43], l'intégrale $P(\mathbf{x}^-, \mathbf{y}^+)$ peut être estimée par :

$$\hat{P}(\mathbf{x}^-, \mathbf{y}^+) = \frac{1}{S} \sum_{s=1}^{S} \frac{P(\mathbf{x}^+_{(s)}, \mathbf{y}^+, \mathbf{x}^-)}{I_{(\mathbf{x}^-, \mathbf{y}^+)}(\mathbf{x}^+_{(s)})} \tag{4.29}$$

où $\mathbf{x}^+_{(1)}, \ldots, \mathbf{x}^+_{(S)}$ sont des échantillons indépendants et identiquement distribués venant de la densité de probabilité $I_{(\mathbf{x}^-, \mathbf{y}^+)}(\mathbf{x}^+)$.

L'estimateur $\hat{P}(\mathbf{x}^-, \mathbf{y}^+)$ est sans biais. Cependant, l'utilisation d'un échantillonnage

préférentiel implique que son coefficient de variation est désormais ([13])

$$c_v[\hat{P}(\mathbf{x}^-,\mathbf{y}^+)] = \frac{\sqrt{\mathrm{Var}\hat{P}(\mathbf{x}^-,\mathbf{y}^+)}}{\mathbb{E}[\hat{P}(\mathbf{x}^-,\mathbf{y}^+)]} = \sqrt{\frac{1}{S}\left(\sum_{\mathbf{x}^+_{(s)}\in\mathcal{X}} \frac{P^2(\mathbf{x}^+_{(s)}|\mathbf{x}^-,\mathbf{y}^+)}{I_{(\mathbf{x}^-,\mathbf{y}^+)}(\mathbf{x}^+_{(s)})} - 1\right)}. \tag{4.30}$$

La principale contrainte concernant le choix de la fonction d'importance concerne le support de $P(\mathbf{x},\mathbf{y}^+)$, qui nécessite d'être inclus dans la fonction d'importance. Cela venant du fait que les fonctions n'incluant pas le support de $P(\mathbf{x},\mathbf{y}^+)$ peuvent amener à des variances infinies. Respectant cette contrainte, toute fonction incluant le support de $P(\mathbf{x},\mathbf{y}^+)$ peut être choisie comme fonction d'importance. Cependant leur efficacité sera complètement différente. Il est alors nécessaire de choisir correctement la fonction d'importance utilisée.

Fonction d'importance optimale La fonction d'importance idéale, notée $I^*_{(\mathbf{x}^-,\mathbf{y}^+)}(\mathbf{x}^+_{(s)})$, est la fonction minimisant la variance de l'estimateur $\hat{P}(\mathbf{x}^-,\mathbf{y}^+)$ et est définie par :

$$I^*_{(\mathbf{x}^-,\mathbf{y}^+)}(\mathbf{x}^+_{(s)}) = P(\mathbf{x}^+|\mathbf{y}^+,\mathbf{x}^-) = \int_{\theta} P(\mathbf{x}^+|\mathbf{y}^+,\mathbf{x}^-,\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathbf{y}^+,\mathbf{x}^-)d\boldsymbol{\theta}. \tag{4.31}$$

Cette intégrale n'étant pas calculable explicitement, nous proposons de l'approcher par la distribution instrumentale Bayésienne 4.32.

$$\boxed{\hat{I}^*_{(\mathbf{x}^-,\mathbf{y}^+)}(\mathbf{x}^+_{(s)}) = \frac{1}{R}\sum_{r=1}^{R} P(\mathbf{x}^+|\mathbf{y}^+,\mathbf{x}^-,\boldsymbol{\theta}_{(r)})}. \tag{4.32}$$

Dans l'équation 4.32 les paramètres $\boldsymbol{\theta}_{(r)}$ sont sélectionnés pour être des réalisations indépendantes de la loi $P(\boldsymbol{\theta}|\mathbf{y}^+,\mathbf{x}^-)$. Afin d'obtenir un bon estimateur $I_{(\mathbf{x}^-,\mathbf{y}^+)}(\mathbf{x}^+_{(s)})$, l'estimation de la loi a posteriori $P(\boldsymbol{\theta}|\mathbf{y}^+,\mathbf{x}^-)$ et la génération des paramètres $\boldsymbol{\theta}_{(r)}$ est effectuée par un échantillonneur de Gibbs.

4.3.4 Échantillonneur de Gibbs

L'échantillonneur de Gibbs [44] est une des méthodes de type Monte Carlo par chaîne de Markov (MCMC) [94],[93]), les plus couramment utilisées. L'objectif de ce type de méthodes est l'utilisation d'une méthode de Monte Carlo pour la simulation

d'une chaîne de Markov ayant la loi recherchée pour loi stationnaire [1].

Description : Le principe de l'échantillonneur de Gibbs est la construction d'une chaîne de Markov à partir de la loi jointe par l'utilisation des loi conditionnelles complètes correspondantes. Ainsi à partir de la loi jointe $P(\mathbf{x}^-, \mathbf{x}^+, \mathbf{y}^-, \mathbf{y}^+, \boldsymbol{\theta})$, l'échantillonneur de Gibbs va générer des chaînes de markov $(\boldsymbol{\theta}^{(t)}, \mathbf{x}^{(t)})_{t=1,\ldots,nbiter}$, t correspondant à l'indice d'un état de la chaîne de Markov, par l'utilisation des lois conditionnelles complètes : $P(\mathbf{x}^+|\mathbf{x}^-, \mathbf{y}^+, \boldsymbol{\theta})$, $P(\mathbf{p}|\mathbf{x}^+, \mathbf{x}^-, \mathbf{y}^+, \mathbf{p}_\cdot)$ et $P(\mathbf{p}_\cdot|\mathbf{x}^+, \mathbf{x}^-, \mathbf{y}^+, \mathbf{p})$, selon les étapes données par algorithme 1. C'est un algorithme récursif qui réajuste itérativement les distributions conditionnelles afin de générer des échantillons aléatoires $\boldsymbol{\theta}^{(t+1)}, \mathbf{x}^{+(t+1)}$, où les distributions conditionnelles sont calculées à partir des distributions jointes. Un trait particulier de l'échantillonneur de Gibbs est que les seules densités utilisées pour les simulations sont les densités conditionnelles complètes.

Les propriétés conjuguées inhérentes à l'inférence bayésienne permettent l'implémentation efficace d'un échantillonneur de Gibbs. L'algorithme 1 présente l'échantillonneur de Gibbs utilisé dans ce travail.

Algorithme 1 : Échantillonneur de Gibbs

1.Initialisation nbiter
2.Initialisation de $\boldsymbol{\theta}^{(0)}$, $\mathbf{x}^{+(0)}$;
3. **for** *t = 1 jusqu'à nbiter* **do**
  2.1. Simulation de $\mathbf{x}^{+(t+1)}$ suivant la loi a posteriori $P(\mathbf{x}^+|\mathbf{x}^-, \mathbf{y}^+, \boldsymbol{\theta}^{(t)})$
  2.2. Simulation de $\mathbf{p}^{(t+1)}$ suivant la loi
  $\mathbf{p}|\mathbf{x}^{+(t+1)}, \mathbf{x}^-, \mathbf{y}^+, \mathbf{p}_\cdot \sim D(n_1^- + n_1^{+(t+1)} + \frac{1}{2}, \ldots, n_p^- + n_p^{+(t+1)} + \frac{1}{2})$
  2.3 Simulation de $\mathbf{p}_\cdot^{(t+1)}$ suivant la loi
  $\mathbf{p}_\cdot|\mathbf{x}^{+(t+1)}, \mathbf{x}^-, \mathbf{y}^+, \mathbf{p} \sim D((n_{11}^- + n_{11}^{+(t+1)} + \frac{1}{2})\delta_{11}, \ldots, (n_{pq}^- + n_{pq}^{+(t+1)} + \frac{1}{2})\delta_{pq})$
end

L'algorithme de l'échantillonneur de Gibbs permet la simulation d'un échantillon de couples $\boldsymbol{\theta}, \mathbf{x}^+$. Deux chaînes de Markov ergodiques $\boldsymbol{\theta}^{(t)}$ et $\mathbf{x}^{+(t)}$, de lois invariantes $P(\boldsymbol{\theta}|\mathbf{x}, \mathbf{y}^+)$, respectivement $P(\mathbf{x}^+|\mathbf{x}^-, \mathbf{y}^+)$, sont ainsi obtenues. Les paramètres $\boldsymbol{\theta}^{(t)}$ simulés par l'échantillonneur de Gibbs étant corrélés entre eux, les réalisations indépendantes $\boldsymbol{\theta}_{(r)}$ sont obtenues en sélectionnant R paramètres $\boldsymbol{\theta}^{(l)}$ toutes les l itérations dans l'échan-

1. La loi stationnaire d'une chaîne de Markov est une loi ne changeant pas au cours du temps, tel que pour la loi π, $\pi = \pi P$, P étant la matrice de transition de la chaîne de Markov

tillon généré par l'algorithme de Gibbs. L'indice l est un pas fixé permettant d'obtenir l'indépendance (ou presque) des données. Les itérations correspondant au temps de "chauffe" ont également été retirées. Le temps de chauffe correspond au temps que met l'algorithme pour atteindre la stationnarité de la loi. Pour éviter de biaiser le résultat, il est préférable de pas prendre en compte les premières estimations.

Ces différentes étapes, récapitulées par l'algorithme 2, permettent donc de calculer le critère **BIL**, donné par

$$\boxed{\mathbf{BIL} = \frac{1}{S}\sum_{s=1}^{S}\frac{P(\mathbf{x}^+_{(s)},\mathbf{y}^+,\mathbf{x}^-)}{\hat{I}^*_{(\mathbf{x}^-,\mathbf{y}^+)}(\mathbf{x}^+_{(s)})}} \tag{4.33}$$

où les $\mathbf{x}^+_{(s)} \sim P(\mathbf{x}^+|\mathbf{x}^-,\mathbf{y}^+,\boldsymbol{\theta}^{(r)})$. Par la suite, la stratégie utilisant la méthode **EXsearch** et

Algorithme 2 : Algorithme pour le calcul du critère BIL

Data : Lecture des données

for *Chaque itération (s) = 1 à S* **do**

1. Gibbs (1)
2. Tirage de R couple $(\boldsymbol{\theta},\mathbf{x}^+)$ avec un pas de l itérations
3. Calcul de

$$\hat{I}^*_{(\mathbf{x}^-,\mathbf{y}^+)}(\mathbf{x}^+_{(s)}) = \frac{1}{R}\sum_{r=1}^{R} P(\mathbf{x}^+|\mathbf{y}^+,\mathbf{x}^-,\boldsymbol{\theta}^{(r)}) \tag{4.34}$$

4. Simulation des $\mathbf{x}^+_{(s)} \sim P(\mathbf{x}^+|\mathbf{x}^-,\mathbf{y}^+,\boldsymbol{\theta}^{(r)})$
5. Calcul de

$$\frac{P(\mathbf{x}^+_{(s)},\mathbf{y}^+,\mathbf{x}^-)}{I_{(\mathbf{x}^-,\mathbf{y}^+)}(\mathbf{x}^+_{(s)})} \tag{4.35}$$

end

7. Calcul de

$$\boxed{\mathbf{BIL} = \frac{1}{S}\sum_{s=1}^{S}\frac{P(\mathbf{x}^+_{(s)},\mathbf{y}^+,\mathbf{x}^-)}{\hat{I}^*_{(\mathbf{x}^-,\mathbf{y}^+)}(\mathbf{x}^+_{(s)})}} \tag{4.36}$$

le critère BIL est appelée EXBIL.

L'objectif des critères BIC et **BIL** est de sélectionner le "quasi-vrai" modèle parmi l'ensemble de modèles $\boldsymbol{\Delta}$. Cependant, la contrainte imposant certaines probabilités de transition à zéro est très forte et restrictive. Afin de sortir de cette contrainte et d'enrichir la famille de modèles, nous proposons d'utiliser les critères de sélection de modèles pour effectuer une agrégation de modèles.

4.4 Bayesian Model Averaging

(BMA) Le Bayesian Model Averaging [55] a pour but d'enrichir notre famille de modèles et sortir de la contrainte possiblement trop forte des zéros. C'est une autre façon d'utiliser les critères de sélection en faisant de l'agrégation de modèle plutôt qu'une sélection de modèle. Un estimateur moyen $\bar{\mathbf{p}}_{\cdot}$ peut alors être obtenu de la façon suivante :

$$\bar{\mathbf{p}}_{\cdot} = \sum_{\delta \in \boldsymbol{\Delta}} \hat{\mathbf{p}}_{\cdot}(\delta) \hat{P}(\delta | \mathbf{x}^{-}, \mathbf{y}^{+}) \tag{4.37}$$

où $\hat{P}(\delta|\mathbf{x}^{-}, \mathbf{y}^{+}) \propto \exp(-\mathrm{BIC}_{\delta})$ lorsque le critère BIC est utilisé et $\hat{P}(\delta|\mathbf{x}^{-}, \mathbf{y}^{+}) \propto \exp(-\mathbf{BIL}_{\delta})$ lorsque c'est le critère **BIL** qui est utilisé. Cette option permet d'obtenir une estimation de $\mathbf{p}_{\cdot}$ permettant de sortir de la contrainte possiblement trop forte des modèles de $\boldsymbol{\Delta}$.

L'objectif du critère **BIL** est de pallier aux différentes limites du critère asymptotique BIC. Pour que le critère **BIL** soit efficace, il est nécessaire d'évaluer la convergence des différents algorithmes en fonction des caractéristiques des jeux de données utilisés. La section suivante présente les performances du critère **BIL**.

4.5 Expériences numériques

Afin d'avoir un algorithme optimal, il est nécessaire de fixer ses paramètres efficacement. Pour l'échantillonneur de Gibbs, cela implique de vérifier que le temps de chauffe ne soit pas pris en compte dans le calcul de l'estimateur final. Les échantillons sur lesquels le critère est appliqué peuvent être très divers, que ce soit en taille d'échantillons, ou de nombre de modalités. Afin de paramétrer l'algorithme correctement selon le jeu de données, l'étude de son évolution est réalisée. Enfin, une comparaison des critères BIC et BIL est réalisée dans le but d'évaluer la performance des deux critères sur des échantillons de petites tailles.

4.5.1 Évolution du Gibbs en fonction de n

L'objectif de cette section est d'évaluer le comportement de l'échantillonneur de Gibbs en fonction de la taille de l'échantillon.

Données Les données sont simulées à l'aide de la fonction Rmultinorm [2], afin de correspondre à un modèle où $p = 2$ et $q = 3$ avec $\mathbf{p} = (0.7, 0.3)$, $\mathbf{p}_1 = (0.7, 0, 0.3)$ et $\mathbf{p}_2 = (0, 0.8, 0.2)$. Dans ces expériences, 5 échantillons de données de tailles différentes $n \in \{70, 500, 1000, 5000, 20000\}$ ont été générés.

Protocole L'algorithme de Gibbs (Algorithme 1) est lancé 1 fois pour chacun des 5 modèles. Pour chaque modèle, le nombre d'itérations *nbiter* du Gibbs a été fixé à 10000. Pour chaque algorithme, les 10 premières itérations, correspondant au temps de chauffe ont été supprimées. Les paramètres $\mathbf{p}$ générés par l'algorithme de Gibbs pour chacun des modèles sont comparés.

La figure 4.4 présente l'évolution de l'échantillonneur de Gibbs pour l'estimation des paramètres $\mathbf{p}$. Il est alors possible de constater que plus l'échantillon généré est grand, plus le paramètre généré par le Gibbs est précis, la qualité de l'estimation est plus précise. De plus, la figure 4.4 permet de constater que quelle que soit la taille de l'échantillon, la loi atteint la stationnarité très rapidement. Seulement 10 itérations suffisent au temps de chauffe. Si un temps de chauffe plus grand était nécessaire à l'algorithme, de plus grandes oscillations sur les premières itérations de l'algorithme seraient constatées sur la figure 4.4.

4.5.2 Évolution du Gibbs en fonction du nombre des modalités p et q

Dans cette section, l'objectif est de constater l'évolution de l'échantillonneur de Gibbs en fonction du nombre de modalités. Pour cela 4 modèles simulés ont été générés avec la fonction Rmultinorm. Chaque échantillon est de taille $n = 5000$, le nombre de modalités pour chacun des modèles est présenté dans le tableau 4.3. Pour chaque modèle, le nombre maximum d'itérations *nbiter* du Gibbs est de 6000. De même que pour les expériences précédentes, les itérations correspondant au temps de chauffe ont été supprimées.

La figure 4.5 montre l'évolution de l'échantillonneur de Gibbs pour l'estimation des paramètres $\mathbf{p}$ des 4 modèles simulés. Il est alors possible de constater que le nombre de modalités ne semble pas influencer la précision des estimations de l'échantillonneur

2. http://stat.ethz.ch/R-manual/R-devel/library/stats/html/Multinom.html

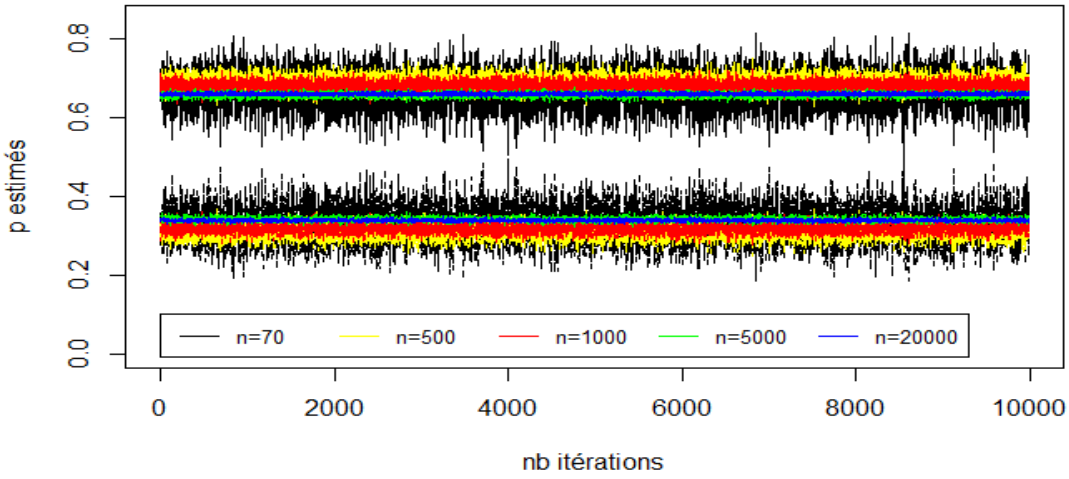


FIGURE 4.4 – Simulation de $\mathbf{p}$ en fonction de n

δ	p	q	$\mathbf{p}$
δ_1	2	3	$\mathbf{p} = (0.8, 0.2)$
δ_2	3	4	$\mathbf{p} = (0.5, 0.3, 0.2)$
δ_3	3	5	$\mathbf{p} = (0.6, 0.2, 0.2)$
δ_4	4	7	$\mathbf{p} = (0.4, 0.3, 0.2, 0.1)$

TABLEAU 4.3 – Caractéristiques des modèles.

de Gibbs. En effet, quel que soit le nombre de modalités du modèle, la précision de l'estimation est similaire.

4.5.3 Convergence du critère BIL en fonction de n

De même que pour l'échantillonneur de Gibbs, afin de paramétrer efficacement l'algorithme et optimiser son temps de calcul, il est nécessaire d'évaluer la convergence du critère **BIL** en fonction de la taille de l'échantillon et du nombre de modalités des différents modèles. Une étude de la convergence du critère a donc été réalisée. Dans cette étude, le modèle simulé est de la forme : $p = 2$ et $q = 3$ avec $\mathbf{p} = (0.7, 0.3)$, $\mathbf{p}_1 = (0.7, 0, 0.3)$ et $\mathbf{p}_2 = (0, 0.8, 0.2)$. Pour étudier la convergence du critère **BIL**, 9 échantillons de données de tailles différentes $n \in \{70, 150, 500, 1000, 2000, 5000, 10000, 15000, 20000\}$ ont été générés. Dans cet algorithme, le nombre d'itérations de l'échantillonneur de Gibbs a été fixé à 500. Afin d'avoir l'indépendance des données le paramètre l a été fixé à 10.

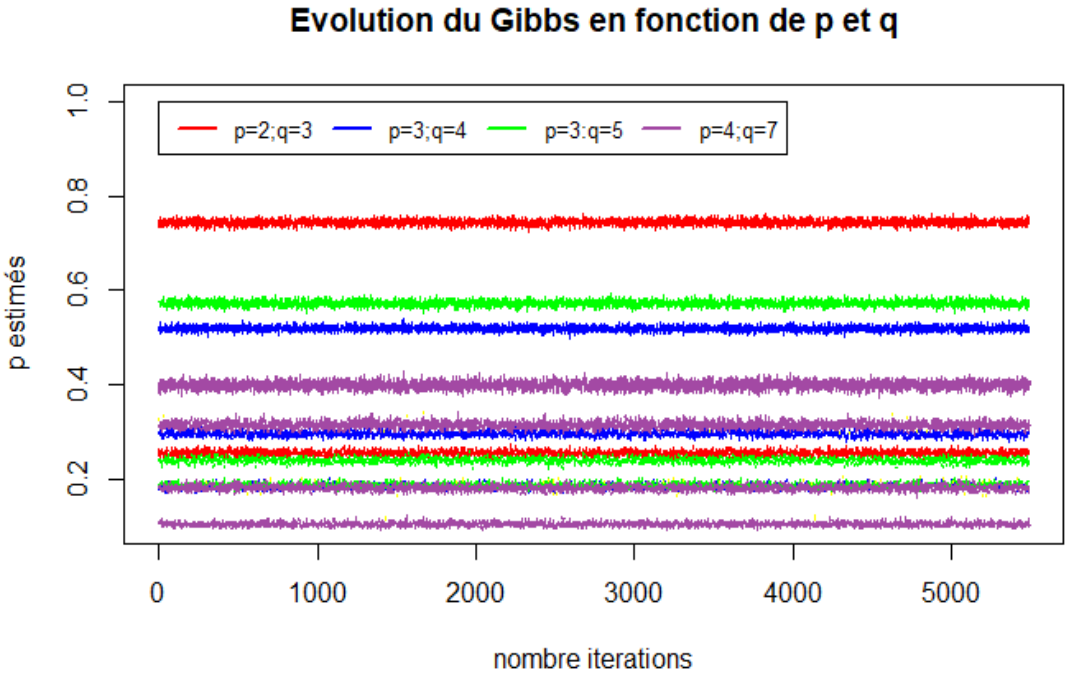


FIGURE 4.5 – Simulation de **p** en fonction du nombre de modalités p, q.

Les R valeurs de paramètres $\theta^{(r)}$ sont sélectionnées parmi les 500 itérations, toutes les 10 itérations, soit $R = 50$. Pour chacun des modèles, le nombre d'itérations de l'échantillonnage préférentiel $S = 5000$.

La figure 4.6 présente la convergence du critère **BIL** en fonction n. D'après cette figure, la convergence du critère **BIL** ne semble pas dépendre de la taille de l'échantillon n. En effet, pour $n = 500$ et $n = 2000$ la convergence semble se faire un peu avant 5000 itérations alors que pour $n \in \{70, 15000\}$ la convergence se réalise aux alentours de 3000 itérations. Cependant, pour chacun des échantillons, le critère **BIL** semble avoir convergé avant 5000 itérations.

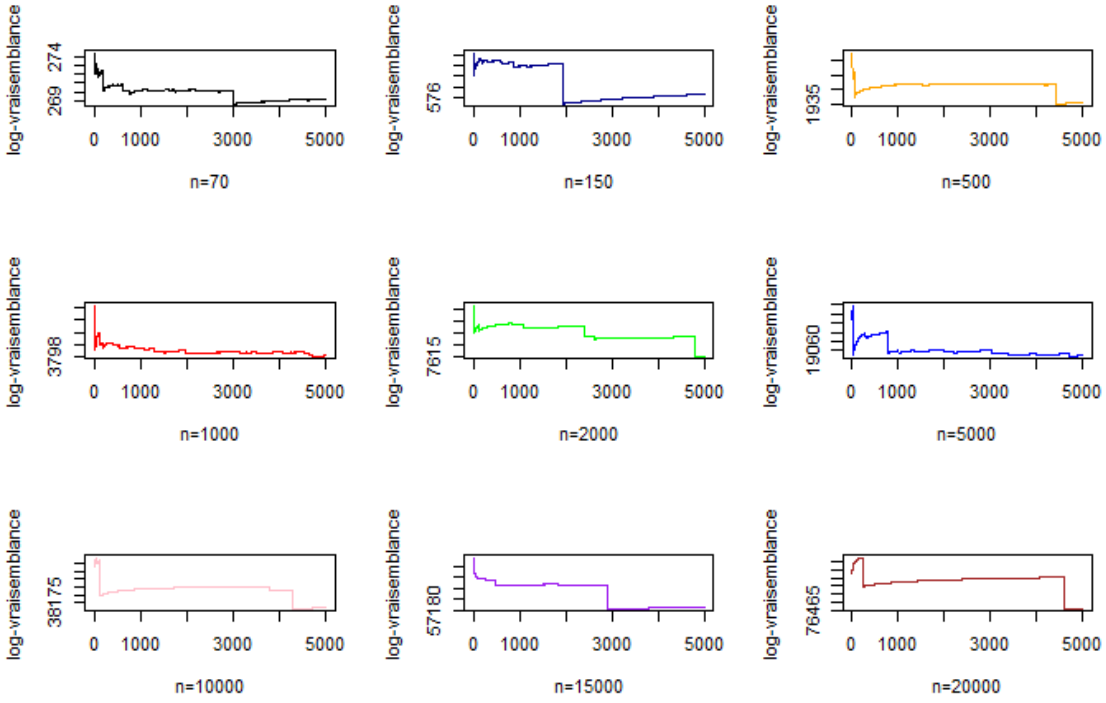


FIGURE 4.6 – Convergence du critère **BIL** en fonction de n.

4.5.4 Convergence du critère BIL en fonction du nombre de modalités p et q

Dans ces expériences, l'objectif est d'étudier la convergence du critère **BIL** en fonction du nombre de modalités des différents modèles. Dans cette étude, les 4 modèles présentés dans le tableau 4.3 sont repris. A l'instar des expériences précédentes, le nombre

d'itérations de l'échantillonneur de Gibbs a été fixé à 500. Les R valeurs de paramètres $\theta^{(r)}$ sont sélectionnées parmi les 500 itérations, toutes les 10 itérations, soit $R = 50$. Pour chacun des modèles, le nombre d'itérations de l'échantillonnage préférentiel est fixé à $S = 2000$.

La figure 4.7 présente la convergence du critère **BIL** en fonction du nombre de modalités p et q. D'après cette figure, la convergence du critère **BIL** ne semble pas être influencée par le nombre de modalités des modèles.

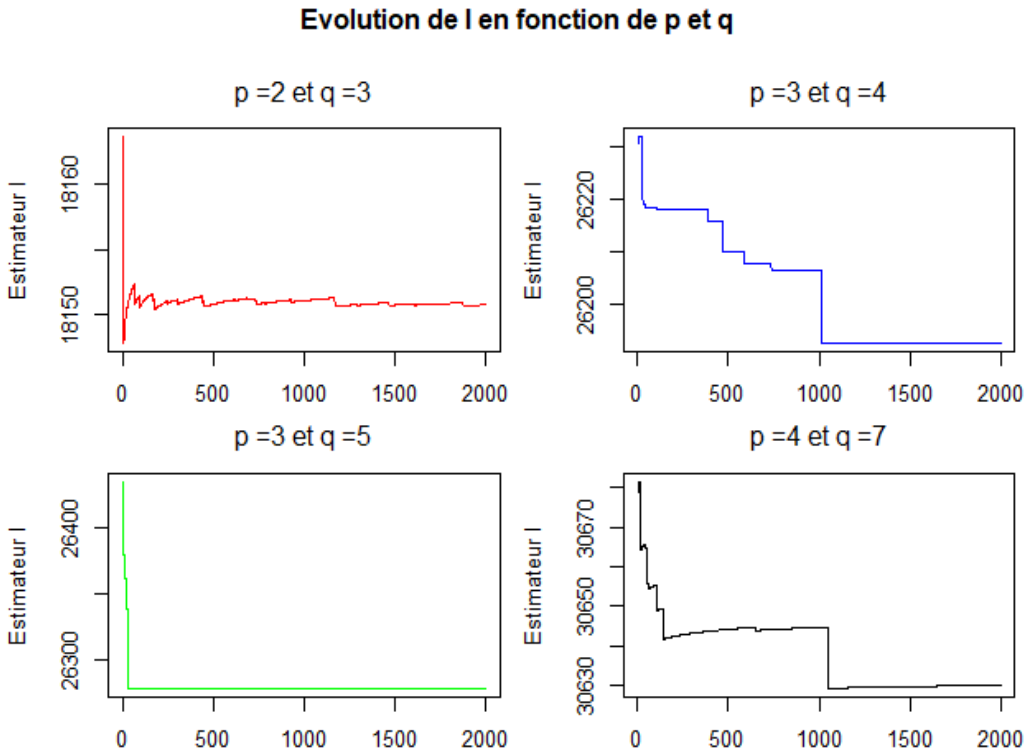


FIGURE 4.7 – Convergence du critère **BIL** en fonction du nombre de modalité p, q.

Afin d'avoir des valeurs de qualité générées par l'échantillonneur de Gibbs, il est préférable d'avoir une taille d'échantillon assez grande. Cependant, la convergence du critère **BIL** ne semble pas dépendre de la taille de l'échantillon, ni du nombre de modalités du modèle. Afin de valider la convergence de l'algorithme, il peut être nécessaire de le lancer à plusieurs reprises.

4.5.5 Comparaison des critères BIC vs BIL

Dans la section 4.2.3, il a été montré que l'approximation asymptotique réalisée dans la construction du critère BIC implique des performances pouvant être mauvaises pour des échantillons de petites tailles. D'autre part, l'approximation de Laplace implique que les performances du critère BIC ne soient théoriquement garanties lorsque les paramètres sont estimés sur le bord de leur espace. Le critère **BIL** a alors été proposé pour pallier à ces limites. L'objectif de cette section est de comparer les performances des critères asymptotiques BIC et non asymptotiques **BIL**.

Taille de l'échantillon

Données & protocoles Dans ces expériences, 100 échantillons ont été générés pour des expériences de différentes tailles $n \in \{20, 40, 70, 100, 150, 200, 300, 500\}$. Tous les échantillons sont générés selon le même modèle où p=2 et q=3 avec $\mathbf{p} = (0.8, 0.2)$, $\mathbf{p}_1 = (0.7, 0, 0.3)$ et $\mathbf{p}_2 = (0, 0.6, 0.4)$. Pour chacune des 8 expériences, 4 modèles sont comparés. Le critère BIC et le critère **BIL** sont calculés pour chacun des 100 échantillons, pour les huit expériences. Pour chaque critère, les modèles ont été classés du modèle minimisant le critère de sélection au modèle le maximisant. Finalement la moyenne du rang des 100 échantillons pour les 8 expériences est réalisée. Le tableau 4.4 présente le rang moyen du modèle simulé pour chacun des critères de sélection de modèles.

Taille de l'échantillon	20	40	70	100	150	200	300	500
BIC	2.47	2.27	1.95	1.86	1.72	1.64	1.43	1.33
BIL	1.67	1.78	1.72	1.68	1.52	1.55	1.38	1.30

Tableau 4.4 – Rang moyen du modèle simulé.

Le tableau 4.4 permet de constater que le rang moyen pour le critère **BIL** est inférieur au rang moyen du critère BIC pour chacun des échantillons. Ce qui signifie que le critère **BIL** retrouve plus fréquemment le modèle simulé que le critère BIC. Le critère BIC est un critère asymptotique dépendant de n. Le tableau 4.4 permet de constater que plus n est grand, plus le rang moyen de critère BIC se rapproche de celui du critère **BIL**. Néanmoins, pour des échantillons de tailles $n \in \{20, 40, 70, 100\}$ le rang moyen pour le critère BIC est supérieur au rang moyen du critère **BIL**. Pour ces tailles d'échantillons, le critère BIC positionne alors fréquemment le modèle simulé en seconde ou troisième position. Il ne retrouve donc pas le vrai modèle. Alors que le critère **BIL**, sur cette même

taille d'échantillon positionne le modèle simulé en moyenne en rang 1. Le critère **BIL** est donc plus efficace que le critère BIC pour retrouver le "quasi-vrai" sur des échantillons de petites tailles. Pour pouvoir remplacer le critère **BIL** par le critère BIC en ayant des performances similaires, la taille de l'échantillon doit être supérieure à 500.

Paramètres estimés sur les bords de son espace

Concernant l'atteinte des bords de l'espace pour les paramètres, les deux critères ont le même comportement. Pour évaluer ce comportement, 100 échantillons de taille 5000 ont été générés, de la forme suivante : p=2 et q=3 avec $\mathbf{p} = (0.7, 0.3)$, $\mathbf{p}_1 = (1, 0, 0)$ et $\mathbf{p}_2 = (0, 0.8, 0.2)$. De même que pour les expériences précédentes, pour chaque critère, les modèles ont été classés du modèle minimisant le critère au modèle le maximisant puis la moyenne a été réalisée. Le rang moyen pour les deux critères est 1. Les deux critères ont retrouvé le modèle simulé pour chacun des 100 échantillons simulés et ont donc des performances similaires sur ce type de modèles.

4.6 Conclusion

Dans le chapitre 3, nous avons proposé une modélisation sous contraintes pour la résolution de notre problème. Les contraintes de cette modélisation consistant à fixer certaines probabilités de transition à zéro. Les probabilités de transition à estimer et celles devant être fixées à zéro étant inconnues, cela implique de travailler avec un ensemble de modèle. L'estimation des probabilités de transition est effectuée par un algorithme EM pour chaque modèle de l'ensemble. Cette méthode étant exhaustive, elle est notée **EXsearch** dans la suite de ce travail. Il est alors nécessaire de comparer les différents modèles de l'ensemble afin de trouver le "meilleur" modèle de l'ensemble. C'est à dire le modèle étant le plus en adéquation avec celui ayant servi à générer les données. Dans ce chapitre, après avoir montré comment détecter deux modèles non identifiables, deux critères de sélection de modèles ont été proposés : un critère asymptotique, le critère BIC, et un critère non asymptotique, le critère **BIL**. De plus, afin d'enrichir la famille de modèles, une stratégie d'agrégation de modèles a également été proposée. Le critère BIC montre de bonnes performances pour la sélection de modèle, cependant, les limites liées à ses approximations sont rapidement atteintes par les modèles proposés et le contexte de ce travail. Un second critère, non asymptotique, a alors été proposé. Ce critère repose sur le calcul de la vraisemblance intégrée des données observées $P(\mathbf{x}^-, \mathbf{y}^+)$. Ce calcul requiert l'utilisation d'une stratégie d'échantillonnage préférentiel et d'un

échantillonneur de Gibbs, rendant son temps d'exécution très long. Il a été montré que le critère BIL est plus performant sur des échantillons de petites tailles ($n < 500$) mais que le critère BIC a des performances similaires sur les autres échantillons. Le calcul du critère **BIL** nécessitant un temps de calcul conséquent afin de réaliser les approximations bayésiennes et ayant un comportement similaire au critère BIC, dans la suite de ce travail, nous proposons deux stratégies. La première stratégie (EXBIC) compare l'ensemble des modèles avec le critère BIC. La seconde stratégie (EXBIL) est d'appliquer le critère **BIL** uniquement sur les 10 premiers modèles classés par le critère BIC, pour les échantillons de taille supérieur à 500. Le but étant de limiter le temps d'exécution de la stratégie, en affinant les résultats du critère BIC sur des modèles ayant des valeurs de critères très proches. Les stratégies EXBIC et EXBIL utilisent une première méthode d'estimation des paramètres pouvant être combinée avec deux critères de sélection de modèles permettant d'effectuer la comparaison de modèles. Les deux stratégies donnent des résultats satisfaisants en terme d'estimation des paramètres et de choix de modèles. Cependant, elles requièrent de réaliser l'estimation des paramètres et la comparaison des modèles pour l'ensemble des modèles. Bien que le calcul du critère BIC soit très rapide suite au calcul de la vraisemblance effectué par l'algorithme EM, la méthode **EXsearch** peut rapidement devenir coûteuse en temps de calcul selon le nombre de modèles présents dans l'ensemble. Dans un contexte industriel, cela peut rapidement devenir problématique. Afin d'optimiser le temps d'exécution de la méthode **EXsearch**, et donc des stratégies EXBIC et EXBIL, dans le chapitre suivant, nous proposons une seconde méthode, basée sur une méta-heuristique, dans le but d'effectuer l'estimation et la comparaison de modèles de façon non-exhaustives.

Chapitre 5

Stratégie de recherche d'un modèle de transfert optimal

L'objectif de ce chapitre est de proposer une seconde méthode non exhaustive, pour l'estimation des paramètres et la sélection de modèles. Cette méthode, basée sur des techniques d'optimisation, a pour objectif d'obtenir des résultats similaires à ceux de la méthode exhaustive, sur les jeux de données simulés et réelles, tout en étant plus rapide. L'intérêt de cette seconde méthode est de pouvoir explorer plus efficacement l'espace de recherche, afin de trouver potentiellement de nouveaux modèles plus en adéquation avec nos données. Dans ce chapitre, nous introduirons dans un premier temps les méthodes d'optimisation mono-objectif stationnaires. Puis, nous présenterons les méta-heuristiques et notamment les méta-heuristiques à base de population. Enfin, nous présenterons la méthode **AGBIC** que nous proposons. Cette méthode a fait l'objet d'une communication lors de la conférence ROADEF 2018 [9], et d'une publication dans la conférence internationale LION12 2018 [10].

5.1 Motivation

La méthode exhaustive **EXsearch**, associée aux critères de sélection BIC et BIL, présentée dans les chapitres 3 et 4 a montré sa capacité à estimer et sélectionner un modèle efficacement. Cependant, la méthode **EXsearch** est exhaustive et très consommatrice en temps, quel que soit le critère de sélection de modèle utilisé. La stratégie **EXBIC** étant plus rapide que la stratégie **EXBIL**, dans cette partie, nous nous focalisons sur l'optimisation de la stratégie **EXBIC**. La méthode **EXsearch** gère deux objectifs. Le pre-

mier objectif est l'estimation de la valeur des paramètres, le second étant la sélection du modèle minimisant la valeur du critère de sélection. Ces deux objectifs peuvent être modélisés comme deux problèmes d'optimisation. Les paramètres étant des probabilités de transition, le premier objectif correspond à un problème d'optimisation continue, alors que la sélection de modèle implique un problème d'optimisation combinatoire. Contrairement à la sélection de modèle, l'estimation des paramètres doit être effectuée pour chaque modèle. Le nombre de problèmes d'optimisation continue est donc équivalent au nombre de modèles comparés. En reprenant le cas d'usage utilisé pour la modélisation probabiliste, où la variable $\boldsymbol{x}$ a 4 modalités et la variable $\boldsymbol{y}$ a 7 modalités, le nombre de modèles à comparer est de $\binom{6*4}{6} = 134\,596$. Dans cet exemple, cela donne 134 596 problèmes continus où les valeurs des probabilités sont à estimer. Suite à cette estimation, l'ensemble des modèles est comparé pour la sélection du meilleur modèle au sens du critère de sélection. Les contraintes imposées sur les modèles, indiquées dans la section 3.2, ont réduit l'espace de recherche, certains modèles n'étant pas autorisés. Malgré ces contraintes, la méthode **EXsearch** reste très longue. A titre d'exemple, dans le cadre du cas d'usage où $p = 4$ et $q = 7$, le temps d'exécution de la stratégie **EXBIC**,implémentée avec le logiciel RStudio, est de 30 minutes. La stratégie est exécutée sur une machine Windows avec un processeur Intel Core i7-4510U CPU 2.00GHz et 16GO de RAM. Afin de réduire le temps d'exécution d' **EXBIC**, les modèles ont été limités à une famille spécifique, où la famille de modèles est définie par :

- Un nombre de paramètres estimés équivalant au nombre de modalités de la variable $\boldsymbol{y} - 1$, soit $q - 1$.
- L'utilisation de probabilités impliquent que la somme des probabilités de transition $\mathbf{p}_{\cdot}$ pour chaque modalité h de la variable $\boldsymbol{x}$ soit égale à 1. Une contrainte supplémentaire est alors ajoutée : la probabilité de transition entre la modalité h et la dernière modalité $h' = q$ de la variable $\boldsymbol{y}$ doit correspondre à $(1-\sum_{h'=1}^{q} \mathbf{p}_h)$. Les probabilités de transition répondant à cette contrainte sont alors fixées comme paramètre non libre. Ce qui signifie que leur valeur ne peut être ni fixée à zéro ni être une valeur estimée.

L'ajout de ces contraintes supplémentaires ne suffit pas à réduire suffisamment le temps d'exécution de la méthode **EXsearch** pour rendre son utilisation optimale dans un contexte industriel. D'autre part, selon la variable modifiée, le nombre de paramètres à estimer et de modèles comparés peut augmenter rapidement. L'objectif est alors de réduire le temps de calcul pour l'estimation et de la comparaison de modèles.

L'utilisation d'une recherche non-exhaustive pour répondre aux différents objectifs semble pertinente. De plus, l'utilisation d'une recherche non-exhaustive aura pour ob-

jectif supplémentaire d'élargir l'espace de recherche avec la suppression de la contrainte imposant à certaines probabilités de transition d'être des paramètres non libres. La section suivante se focalise sur ces méthodes. Dans un premier temps, les méthodes non-exhaustives existantes sont introduites. Dans un second temps, nous détaillerons les méta-heuristiques et plus particulièrement les méta-heuristiques à base de population. Finalement, nous présenterons la méthode proposée AGBIC, basée sur une méta-heuristique à base de population pour résoudre le problème d'estimation et de sélection de modèles.

5.2 Etat de l'art

5.2.1 Méthodes d'optimisation

De nombreuses méthodes d'optimisation permettent de trouver la solution optimale à un problème. Selon la complexité du problème, ces méthodes peuvent être classées en deux catégories :

Méthodes exactes Les méthodes exactes se caractérisent par l'obtention de la solution optimale et garantissent cette optimalité. Elles sont généralement basées sur une recherche exhaustive dans l'espace de recherche afin de trouver la solution optimale. Ce qui implique une limitation de ces méthodes aux problèmes de petites et moyennes tailles afin d'éviter l'explosion combinatoire et donc l'explosion du temps de calcul.

Méthodes approchées Les méthodes approchées se caractérisent par la génération de solutions de haute qualité dans un temps raisonnable. Cependant, ces méthodes ne possèdent pas de garanties théoriques quant à l'optimalité des résultats obtenus.

Le cadre des méthodes approchées semble le plus approprié pour l'atteinte de nos objectifs. La figure 5.1 permet de constater que les méthodes approchées peuvent être également distinguées en deux catégories : les heuristiques spécifiques et les méta-heuristiques. Ces deux catégories se différencient par leurs champs d'applications. Les heuristiques spécifiques sont des méthodes de résolutions spécifiques à un problème donné. A l'inverse, les méta-heuristiques sont des algorithmes universels pouvant être appliqués pour résoudre la plupart des problèmes d'optimisations [106]. Dans ce travail, nous choisissons d'utiliser les méta-heuristiques pour la résolution de notre problème. Les principales raisons étant la simplicité et la rapidité de mise en place des méta-

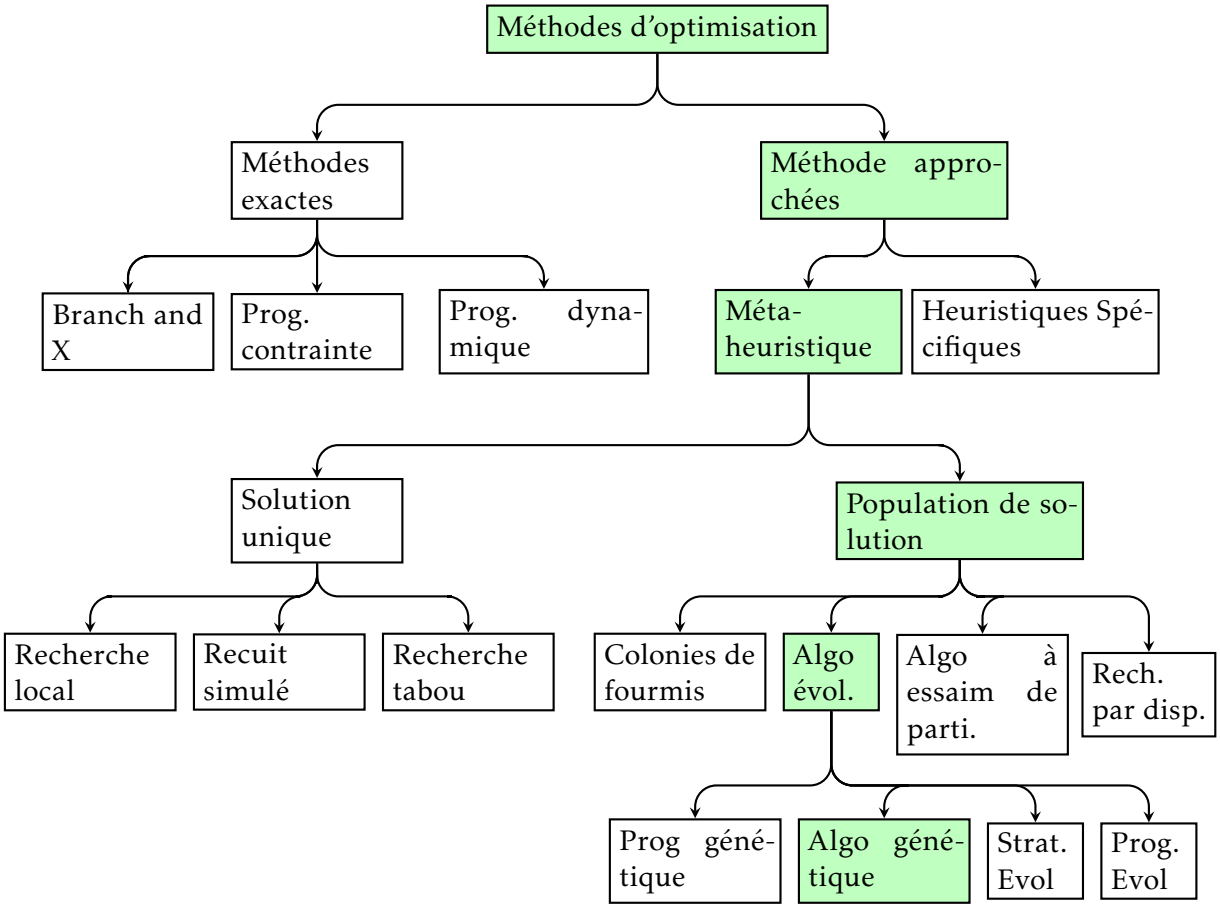


FIGURE 5.1 – Méthodes d'optimisation, en vert, l'approche utilisée dans ce travail.

heuristiques ainsi que leur capacité d'application à différents problèmes. Elles sont détaillées dans la partie suivante.

5.2.2 Méta-heuristiques

Les méta-heuristiques font partie des algorithmes stochastiques approchés. Ces algorithmes permettent de trouver une solution satisfaisante en un temps raisonnable pour des problèmes de grandes dimensions ou des problèmes d'optimisation difficiles.

Deux objectifs sont à distinguer :

Exploration L'objectif est l'exploration de l'espace de recherche.

Intensification L'objectif est l'exploitation de la meilleure solution trouvée.

Le cadre de ce travail et l'objectif de la comparaison de modèle impliquent de se placer dans le cadre exploratoire.

Un grand nombre de méta-heuristiques existe, pouvant être classé de différentes façons [106]. Nous choisissons d'utiliser la classification séparant les méta-heuristiques en deux catégories selon leur base de solutions : recherche basée sur une seule solution, recherche basée sur une population de solutions (cf. Figure 5.1).

Recherche basée sur une seule solution : Le principe des algorithmes basés sur une seule solution est la manipulation et la transformation d'une seule solution pendant la recherche. L'objectif de ces méta-heuristiques est plutôt orienté vers l'intensification car elles permettent l'intensification la recherche dans une région locale. Parmi les algorithmes les plus connus, les Méthodes de descentes [83], le Recuit Simulé [64], la Recherche Tabou [45],[51] ou encore la Recherche Locale Itérée [72], [77] peuvent être cités.

Recherche basée sur une population de solutions : Contrairement aux algorithmes basés sur une seule solution, les algorithmes basés sur une population de solutions, impliquent l'ensemble des solutions de la population dans la recherche. L'objectif de ces méta-heuristiques est plutôt exploratoire, ces méta-heuristiques permettant une meilleure diversification dans l'ensemble de l'espace de recherche. Parmi les méthodes les plus répandues appartenant à cette catégorie : la recherche par dispersion [46], les colonies de fourmis [33], les essaims de particules [61], ou encore les algorithmes évolutionnaires [106] peuvent être citées.

L'un de nos objectifs est la comparaison d'un ensemble de modèles, dans le but de trouver le modèle le plus en adéquation avec nos données. Un ensemble de solutions est alors à explorer et les méta-heuristiques où la recherche implique une population de solutions semblent alors adaptées aux problèmes de sélection de modèle. Cette catégorie de méta-heuristiques comprend plusieurs types d'algorithmes exploitables.

5.2.3 Méta-heuristiques à base de population de solutions

Les méta-heuristiques à base de population (P-Méta-heuristiques) peuvent être vues comme une amélioration itérative d'une population de solutions. L'intérêt de ce type de méta-heuristique est l'utilisation de la population comme facteur de diversité.

Principe Dans un premier temps, une initialisation de la population est réalisée, puis une nouvelle population de solution est générée. Cette nouvelle population est ensuite intégrée à la population courante par l'utilisation de procédures de sélection. Le processus de recherche est arrêté lorsque le critère d'arrêt est satisfait. Généralement,

trois grands groupes de méthodes sont distingués : La recherche par dispersion, les algorithmes basés sur l'intelligence en essaims qui reprennent les méthodes de colonies de fourmis et les méthodes d'optimisation à essaims de particules et enfin les algorithmes évolutionnaires.

Recherche par dispersion (Scatter Search) : La recherche par dispersion a été proposée par Glover en 1977 [46]. C'est une stratégie déterministe qui a été appliquée avec succès sur de nombreux problèmes combinatoires et continus. Le principe de cette méthode est la génération de nouvelles solutions à partir d'un ensemble de solutions dit de "référence". Les nouvelles solutions sont créées par combinaison des solutions de l'ensemble de références. Dans un premier temps, une population initiale est générée. Dans un second temps, le jeu de référence est construit par sélection, au sein de la population initiale, de solutions représentatives. Une solution est ensuite créée par combinaison des solutions sélectionnées. La solution provisoire est ensuite transformée, dans le but d'être améliorée par une méthode reposant généralement sur une méta-heuristique à solution unique. L'ajout des meilleures solutions trouvées permet la mise à jour de l'ensemble de référence. L'évaluation des solutions est finalement effectuée selon leur qualité ou leur diversité. L'algorithme est itéré jusqu'à un critère d'arrêt défini. Il est à noter que pour cette méthode, l'échantillon de référence est de taille moyenne et que le processus de combinaison n'est pas aléatoire mais guidé.

Colonie de fourmis : Le principe de cette méthode, proposée par Dorigo [33] [32], est d'imiter le comportement réel des fourmis cherchant leur nourriture pour résoudre des problèmes d'optimisations, notamment de recherche de plus court chemin. Dans un premier temps, les fourmis explorent de manières aléatoires les environs de leur nid afin de trouver une source de nourriture. Une fois la source de nourriture trouvée, les fourmis reviennent vers leur nid en laissant une trace chimique dite "phéromone", qui est odorante afin d'être détectable, le but étant de retrouver le chemin par la suite. Les chemins avec la plus forte concentration de phéromones sont généralement choisis par les fourmis. Comme les phéromones s'évaporent au cours du temps, les chemins les plus longs seront alors moins détectables et les chemins les plus courts seront les plus choisis. Il est à noter que le plus court chemin devient détectable lorsque les trajets de phéromones sont empruntés par une colonie de fourmis. D'un point de vue optimisation combinatoire, l'objectif est alors de travailler avec une population de solutions. Chaque fourmi choisi un trajet et trace un chemin. L'ensemble de la

population parcourt un certain nombre de trajets, chaque fourmi pondérant le trajet par une quantité de phéromones proportionnelle à la qualité du parcours. La solution est alors évaluée à travers une fonction objective et les traces les plus faibles sont supprimées. Cette méthode a été employée avec succès sur de nombreux problèmes dont le problème du voyageur de commerce. Cette méthode ayant surtout un comportement exploratoire, elle est souvent combinée à des recherches locales pour de meilleures performances.

Algorithmes à essaims de particules (PSO) : A l'instar des colonies de fourmis, les algorithmes à essaim de particules sont inspirés de la nature. Introduites par Kennedy [61], ces méthodes sont inspirées du comportement collectif qu'ont certains animaux tels que les essaims d'insectes, les bancs de poissons ou les nuées d'oiseaux. Le concept de ces méthodes est d'imiter le déplacement groupé et coordonné qu'ont ces animaux pour arriver à trouver de la nourriture ou éviter un obstacle. Ce comportement peut s'apparenter à un problème d'optimisation. Dans l'algorithme, la population est appelée essaim et les individus sont des particules. Une particule est définie par une position initiale et une vitesse de déplacement. Une solution candidate du problème est représentée par une particule. Le déplacement de la particule s'effectue selon sa meilleure position (gardée en mémoire), sa vitesse de déplacement et la meilleure position du voisinage de la particule. En fonction de ces éléments, la meilleure performance connue de la particule est choisie, et sa vitesse est modifiée selon la meilleure performance connue de son voisinage et de ses propres informations.

Algorithme évolutionnaire (EA) : Les algorithmes évolutionnaires sont inspirés de la théorie de l'évolution proposée par C. Darwin [27] en 1859. Ce sont des P-Meta-heuristiques stochastiques qui ont été appliquées avec succès à de nombreux problèmes réels et complexes (multiobjectifs, problèmes hautement contraint/ multimodaux,...). Leurs succès viennent du fait que ces algorithmes peuvent résoudre de nombreux types de problèmes, notamment continus et combinatoires. Les algorithmes évolutionnaires sont basés sur une notion de compétition. Le principe de ces algorithmes repose sur la simulation de l'évolution d'une espèce tel que le montre la figure 5.2. Partant d'une population initiale aléatoire, deux solutions, dites "parents", sont sélectionnées. Puis des solutions, dites "enfants", sont générées à partir des solutions "parents". La génération des solutions "enfants" s'effectue à travers différents opérateurs tel que le croisement ou la mutation. L'étape de génération des enfants est appelée la reproduction. Les "enfants" sont

ensuite évalués par une fonction objective. Si les "enfants" sont performants et capables de survivre dans la population alors ils sont intégrés à la population et remplacent les "parents".

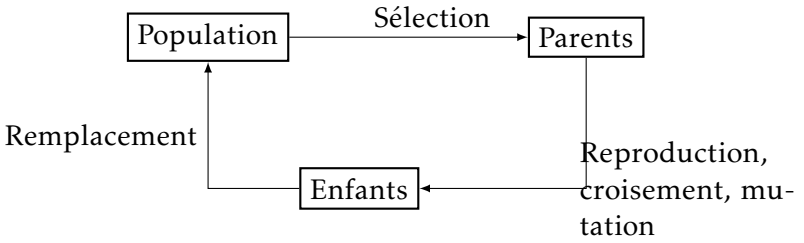


FIGURE 5.2 – Principe d'un algorithme évolutionnaire.

Plusieurs approches ont été proposées telles que les stratégies d'évolution [91], [101], la programmation évolutionnaires [38], les algorithmes génétiques [56] ou encore la programmation génétique [65].

Stratégie d'évolution Les stratégies d'évolutions ont été initialement développées par Rechenberg [91] et Schewefel [101] en 1964 à Berlin. Elles sont principalement appliquées sur des problèmes d'optimisations continus et basées sur des vecteurs de valeurs réelles. Le principe des stratégies d'évolution est la génération d'une population enfant de taille λ, provenant d'une population parent de taille μ ayant subi une mutation. La mutation utilisée dans ces stratégies est une mutation spécifique basée sur une distribution gaussienne. Cette mutation s'effectue par l'ajout d'une valeur aléatoire provenant d'une distribution gaussienne aux valeurs réelles du codage. Suite à la mutation, un opérateur de remplacement élitiste est utilisé afin de sélectionner les enfants survivants. Dans ces stratégies, les opérateurs de croisement ou recombinaison sont rarement utilisés. Il est également important de noter qu'une distinction est effectuée entre la taille de la population des parents μ et la taille de la population des enfants λ tel que $\lambda \leq \mu$.

Programmation évolutionnaire Introduite par Fogel (1962)[38] la programmation évolutionnaire est assez similaire aux stratégies d'évolutions et est donc moins utilisée que les autres familles d'algorithmes évolutionnaires. La programmation évolutionnaire met l'accent sur la mutation et n'utilise pas d'opérateur de recombinaison ou de croisement. De plus, pour ces méthodes, aucune contrainte sur la représentation n'est requise. Concernant le choix des parents, celui-ci est déterministe. La sélection des enfants survivants (rem-

placement) est réalisée de façon probabiliste et est basée sur un tournoi stochastique (Eiben et Smith,2003)[35].

Algorithme génétique Parmi les méthodes évolutionnaires, les algorithmes les plus populaires sont les algorithmes génétiques. Holland [56] fût le premier à les introduire. Les algorithmes génétiques sont inspirés des mécanismes biologiques à l'instar les lois de Mendel et reposent sur le principe fondamental de la sélection naturelle de Charles Darwin [26]. Le principe initial de ces algorithmes, proposé par Holland, est de s'inspirer des principes de la génétique pour les appliquer aux programmes informatiques. Le but étant l'imitation de l'évolution des êtres vivants par les programmes informatiques lors de la recherche de solutions à un problème. Dans un premier temps, les principes fondamentaux des algorithmes génétiques sont formalisés par Holland. L'utilisation des processus de mutation et de croisement génétiques sont également introduit pour présenter l'ajout d'intelligence dans un programme. Dans un second temps, Goldberg reprend la théorie des algorithmes génétiques en ajoutant les parallèles suivant [47], [60] :

— Un individu est lié à un environnement par son code d'ADN.
— Une solution est liée à un problème par son indice de qualité.
— Une "bonne" solution à un problème donné peut être vue comme un individu susceptible de survivre dans un environnement donné.

Le principe général des algorithmes génétiques est la simulation du processus d'évolution d'une population. Dans un algorithme génétique, un individu correspond à une solution du problème et un ensemble de solutions correspond à une population. Partant de cette population, des opérateurs simulant les principes de la génétique telle que le croisement ou la mutation interviennent. La survie des individus dépend ensuite de leur capacité d'adaptation à un environnement. Finalement, une population de solutions de plus en plus adaptée au problème est créée. L'évaluation de cette adaptation est réalisée par une fonction objective.

Il est à noter, que contrairement à la recherche par dispersion, la population initiale de solution peut être de grande taille. De plus, contrairement aux stratégies d'évolutions, l'algorithme génétique n'impose pas une représentation des solutions codées avec des valeurs réelles.

Ce travail comporte un problème pouvant être modélisé est à la fois de façon continu et combinatoire. D'autre part, la population de solutions peut se révéler assez grande. De ce fait, l'utilisation des algorithmes génétiques pour

la résolution de notre problème paraît pertinente. Nous choisissons donc de les utiliser pour l'estimation et la sélection de modèles. Ils seront alors plus particulièrement détaillés dans la section 5.4.2.

Programmation génétique La Programmation Génétique est l'approche évolutionnaire la plus récente. Proposée par J. Koza [65], elle est considérée comme une approche spécialisée des algorithmes génétiques où la population est constituée de programmes informatiques. La différence majeure entre les algorithmes génétiques et la programmation génétique est la représentation des individus. En programmation génétique, les individus sont des programmes informatiques représentés généralement sous forme d'arbres non linéaires. Le principe de la programmation génétique est de faire évoluer une population constituée d'un grand nombre de programmes. L'initialisation de la population initiale, l'évaluation des solutions ainsi que les opérateurs de croisement et de mutation sont appliqués de la même façon que pour les algorithmes génétiques. Cependant, les opérateurs de croisement et de mutation sont adaptés à la représentation des solutions utilisées et travaillent directement sur la structure en arbre des programmes. L'évaluation des solutions est réalisée par une méthode propre au problème posé, les meilleures solutions sont ensuite sélectionnées afin de former la population suivante ayant une chance plus élevée de survivre ou d'avoir des enfants.

5.2.4 Algorithme génétique

Dans le cadre de ce travail, nous faisons le choix d'utiliser un algorithme génétique car il s'adapte particulièrement à nos objectifs. Ces principaux atouts étant : sa capacité à obtenir une solution de "bonne" qualité rapidement, sa capacité à travailler avec de grandes populations et sa flexibilité. Dans un premier temps, nous détaillons les principes de cet algorithme.

L'algorithme génétique est inspiré de phénomènes apparentés à la génétique, le but étant d'imiter le processus d'évolution des êtres vivants. De ce fait, la terminologie de la génétique s'applique également à l'algorithme. Dans l'algorithme génétique, un individu est appelé **chromosome**, et la population est un **ensemble de chromosomes**. A l'instar du processus génétique, l'évolution de ces chromosomes est réalisée par combinaison de ceux-ci. Cette étape est l'étape de **reproduction**. La reproduction pouvant être réalisée par **mutation** ou **croisement** des chromosomes.

Dans la théorie de l'évolution, les individus les plus robustes sont ceux s'adaptant le mieux à leur environnement. L'algorithme génétique applique ce principe pour l'évolution de sa population. Les chromosomes sont alors évalués selon leur capacité d'adaptation. La capacité d'adaptation des chromosomes est évaluée par un **indice de qualité (fitness)**, l'indice de qualité étant calculé par une **fonction d'évaluation**. La figure 5.3 présente le fonctionnement itératif de l'algorithme.

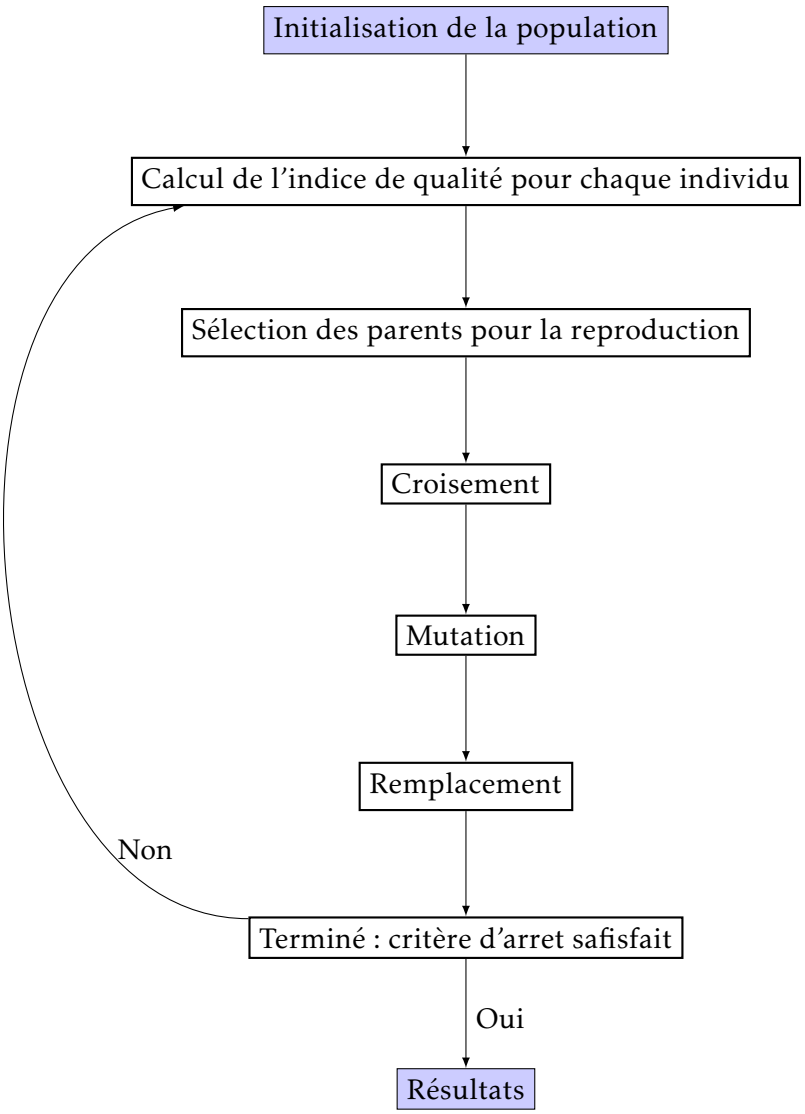


FIGURE 5.3 – Principe d'un algorithme génétique basique.

Le développement d'un algorithme génétique implique de déterminer les éléments suivant :

— La représentation utilisée pour les solutions (le codage du chromosome)
— La fonction d'évaluation
— Les opérateurs utilisés
— Les paramètres de l'algorithme

Représentation Pour que l'algorithme puisse trouver l'ensemble des solutions, il est nécessaire que la représentation choisie permette de coder l'ensemble des solutions du problème. Une solution non représentée étant introuvable pour l'algorithme.

Fonction d'évaluation Cette fonction a pour objectif l'évaluation de chaque chromosome selon le problème donné. La sélection des chromosomes pour l'étape de reproduction est ensuite effectuée selon leur évaluation. Les chromosomes les mieux évalués ont le plus de chance d'être sélectionnés pour l'étape reproduction et sont ceux ayant le plus de chance de transmettre les meilleurs gênes. Le choix de la fonction d'évaluation est important car un mauvais choix de fonction peut impliquer une mauvaise évaluation des chromosomes. De ce fait, des chromosomes de faibles qualités pourraient être sélectionnés pour l'étape de reproduction et induire en erreur l'algorithme dans la recherche de la meilleure solution.

Opérateurs

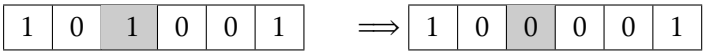
Un algorithme génétique requiert quatre types d'opérateurs : initialisation, sélection, croisement et mutation. Ces opérateurs agissent directement sur les individus de la population. Plus spécifiquement, les opérateurs de croisement et de mutation permettent l'évolution des individus et la création de nouvelles solutions. Il est à noter que les opérateurs existants sont dépendants de la représentation choisie.

Initialisation L'opérateur d'initialisation est utilisé pour la génération de la solution initiale de l'algorithme. Le but étant d'avoir une représentation diversifiée, le plus souvent, la génération des chromosomes est aléatoire.

Sélection La sélection permet l'obtention d'une plus grande proportion de solutions de bonnes qualités par rapport aux solutions de mauvaises qualités. La sélection est réalisée par une heuristique où les bonnes solutions sont supposées être les plus prometteuses pour la génération future. Il existe différentes méthodes de sélection dont la sélection en tournoi qui est présentée plus en détail dans la section 5.3.2

Croisement L'opérateur de croisement consiste à croiser les gènes d'un ou plusieurs parents avec pour objectif, l'obtention d'un ou plusieurs enfants. Différents types de croisements existent, dont le croisement uniforme dont le principe est d'échanger chaque paramètre selon une probabilité fixée à $\frac{1}{2}$. Il est également présenté en section 5.3.2

Mutation L'objectif de la mutation est l'introduction d'une certaine diversité, par la génération de points dans des régions qui sembleraient au préalable sans intérêt [114]. La mutation permet donc l'exploration efficace de l'espace de recherche. Le principe de cet opérateur est de faire apparaître de nouveaux gènes. La mutation la plus simple consiste à changer un paramètre de façon aléatoire en fonction du taux de probabilité de mutation. Chaque chromosome ayant une probabilité de mutation correspond à un taux α. Par exemple :



Remplacement L'étape de remplacement correspond à la sélection des survivants venant de la population des parents et de la population des enfants. La taille de la population étant constante, il est possible de retirer des individus afin d'en ajouter de nouveaux selon une stratégie donnée. Deux grandes stratégies de remplacement existent.

Remplacement générationnel L'objectif du remplacement générationnel est de remplacer l'ancienne génération par la nouvelle. Il concerne donc l'ensemble de la population de taille μ où l'ensemble de la population des enfants remplace systématiquement la population des parents.

Remplacement Steady-state [113] Contrairement au remplacement générationnel, le remplacement steady-state consiste à générer seulement un ou deux enfants à chaque génération. Les enfants générés remplacent alors les pires individus de la population des parents.

Pour ces deux stratégies, plusieurs schémas distincts peuvent être appliqués. Notamment l'élitisme, consistant à toujours sélectionner les meilleurs individus venant de la population des parents et de la population des enfants. L'élitisme mène à une convergence rapide, mais amène parfois également à une convergence prématurée [106].

5.3 Un algorithme pour l'estimation de paramètre et la sélection de modèle : AGBIC

Dans cette partie, nous présentons la méthode proposée pour réaliser l'estimation des paramètres et la sélection de modèles. La méthode proposée, notée (**AGsearch**) est basée sur un algorithme génétique, dont l'utilisation pour résoudre un problème de sélection de modèle a déjà fait ses preuves dans la littérature. Dans la méthode (**AGsearch**), l'algorithme génétique a été adapté pour répondre aux différentes contraintes imposées dans la modélisation réalisée dans le chapitre 3. Dans ce chapitre, nous présentons la stratégie AGBIC, qui correspond à l'utilisation de la méthode **AGsearch** utilisant le critère BIC pour la sélection de modèles.

5.3.1 Etat de l'art

Dans la littérature, de nombreux travaux existent, où le domaine de la statistique et de l'optimisation sont liés. Plus spécifiquement, plusieurs travaux, dont [84], [116], [15], [7], [14] utilisent des algorithmes génétiques pour réaliser une sélection de modèles. Les conclusions de ces travaux montrent que l'utilisation d'un algorithme génétique pour réaliser de la sélection est pertinente. Par exemple, les auteurs de [84] utilisent un algorithme génétique pour faire de la sélection de modèles en régression. Dans ces travaux, deux algorithmes génétiques sont proposés, l'un permettant la sélection des variables à inclure dans le modèle et le second permettant de trouver, en plus des variables, les transformations mathématiques nécessaire à l'obtention d'un modèle ayant des performances optimales. Les modèles proposés par les deux algorithmes génétiques (AG) sont évalués selon trois critères de sélection de modèles : AIC, BIC et SIC (Schwarz Information Criterion) et sont comparés à la sélection Stepwise. Selon ces critères, les algorithmes permettent la découverte de nouveaux modèles comparés à la sélection Stepwise avec des résultats supérieurs à la sélection stepwise en termes de critère de sélection de modèles. Pigeot et Blauth [15] utilisent également un algorithme génétique pour réaliser une sélection de modèles. Dans ces travaux, un modèle est représenté sous forme de graphe orienté où l'objectif est de parcourir l'espace des sous graphes possibles pour retrouver le sous graphe correspondant au modèle ayant permis de générer les données. Les performances de l'algorithme génétique sont ensuite comparées à des méthodes de sélection stepwise et forward où les arcs sont supprimés ou ajoutés itérativement. Les auteurs montrent alors que l'algorithme génétique permet de trouver le "vrai" modèle ou des modèles très proches. Les modèles sont évalués selon le critère

BIC. Les auteurs de [7] comparent également l'AG aux critères d'information statistique tel que le critère BIC ou AIC pour réaliser la sélection de modèles. De même que pour les papiers précédents l'AG montre de bonne performance pour la sélection de modèles, notamment dans de grands espaces de recherches. D'autre part, les auteurs [14] utilisent l'algorithme génétique pour faire de la sélection de modèles. Le but étant de trouver les meilleurs paramétrages d'un modèle non linéaire à effet mixte.

L'algorithme génétique est donc particulièrement utilisé pour faire de la sélection de modèles et a montré, à de multiples reprises de bonnes performances. Il est à noter que dans chacun de ces travaux, la représentation choisie est une représentation binaire où le but n'est pas de faire de l'estimation de paramètre mais une sélection de paramètres permettant l'atteinte de meilleurs modèles.

5.3.2 AGBIC

Représentation et évaluation

Un algorithme génétique est défini par : une solution potentielle, une population, un environnement (espace de recherche) et une fonction d'évaluation. La configuration de l'algorithme génétique utilisé dans la méthode AGBIC repose sur la modélisation probabiliste réalisée dans le chapitre 3. Cela implique l'utilisation de l'ensemble des modèles contraints ayant des probabilités de transition fixées à zéro et du critère de sélection BIC présenté dans la section 4.2.2.

Solution Les modèles correspondant à la modélisation réalisée dans le chapitre 3 sont composés de probabilités de transition fixées à zéro et de probabilités de transition estimées. Une solution correspond à un modèle noté δ de ce type. La Figure 5.4(a) montre un exemple de solutions δ et la Figure 5.4 (b) correspond à sa forme matricielle.

Population Les contraintes imposées dans la section 3.3 impliquent un ensemble de modèles $\mathbf{\Delta}$. La population de l'algorithme est composée de cet ensemble de modèles. De plus, contrairement à la méthode **EXsearch**, le jeu de modèles $\mathbf{\Delta}$ ne doit pas être réduit.

Représentation Les modèles définis dans les chapitres 3 et 4 sont composés de probabilités de transition et de paramètres fixés à zéros. Les éléments du vecteur $\mathbf{p}_{\cdot}$, définis dans la section 3.2.2, sont les probabilités de transition estimées pour un modèle δ. Les propriétés des probabilités impliquent que leurs valeurs soient comprises dans l'intervalle $[0,1]$. L'un des objectifs de l'algorithme étant l'esti-

mation de ces probabilités de transition, leur représentation est alors effectuée par un codage réel. La Figure 5.4 (b) montre un exemple de représentation de la solution avec les paramètres fixés à zéro et les probabilités de transition estimées.

Fonction d'évaluation Une solution correspond à un modèle probabiliste dont l'évaluation est réalisée par un critère de sélection de modèles, présenté dans la section 4.2. Pour pouvoir comparer les deux méthodes, le même critère de sélection de modèles doit alors être utilisé. Dans AGBIC, l'évaluation de la solution est réalisée à travers le critère de sélection de modèle asymptotique BIC, défini dans la section 4.2.2 et par l'équation 5.1 :

$$\boxed{BIC_\delta = -2\ell_\delta(\hat{\boldsymbol{\theta}}; \mathbf{x}^-, \mathbf{y}^+) + \nu_\delta \ln(n)} \,. \tag{5.1}$$

A l'instar de la stratégie EXBIC, l'objectif est alors la minimisation de ce critère.

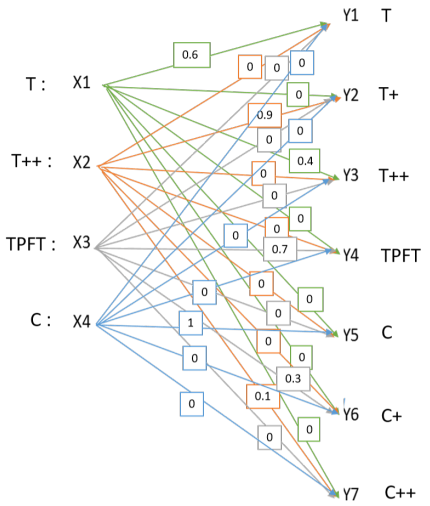
Stratégie de remplacement L'un des objectifs de l'algorithme est d'obtenir une "bonne" solution rapidement. Dans ce but, une stratégie de remplacement steady-state [113] est utilisée. Cette stratégie est combinée à un schéma élitiste afin d'avoir une convergence plus rapide.

5.3.3 Opérateurs

Un algorithme génétique repose sur 3 opérateurs qui sont : la sélection, le croisement, et la mutation. La sélection sera effectuée par un opérateur classique, la sélection par tournoi binaire [79]. Pour ce problème, le codage choisi est un codage réel, les opérateurs de croisements et de mutation seront choisis en conséquence. De plus, pour effectuer le croisement et la mutation, ces opérateurs sont à adapter à notre problème. Les modèles comportant de nombreux zéros, la méthode exhaustive montre que la place des zéros est une réelle information. Le premier objectif sera alors de créer des opérateurs adaptés afin de gérer l'information des zéros.

Opérateurs de sélection

La sélection sera effectuée avec un opérateur classique : la sélection par tournoi binaire [79]. Cet opérateur sélectionne aléatoirement deux solutions de la population $\boldsymbol{\Delta}$. Dans un second temps, la solution est évaluée avec le critère BIC_δ et la meilleure solution $\hat{\delta}$ est sélectionnée. La figure 5.5 montre un exemple de sélection par tournoi où l'individu minimisant la valeur de BIC est sélectionnée par l'opérateur :



(a) Graphe des appariements possibles entre X- variable avant la modification et Y- variable après la modification pour un modèle δ fixé.

$X \backslash Y$	T	T+	T++	I	TR	TR+	TR++
T	0.6	0	0.4	0	0	0.	0
T++	0	0.9	0	0	0	0	0.1
I	0	0	0	0.7	0	0.3	0
TR	0	0	0	0	1	0	0

(b) Matrice de transition correspondant au modèle δ entre les modalités de X and Y.

FIGURE 5.4 – Exemple de représentation de la solution.

$$\hat{\delta} = \operatorname{argmin} \mathrm{BIC}_{\delta} \,. \tag{5.2}$$

Opérateurs de croisements

Concernant les opérateurs de croisements, nous en proposons deux qui seront comparés par la suite : le croisement uniforme et le croisement binaire simulé, afin de garder le plus efficace.

Croisement uniforme [92] : L'opérateur de croisement uniforme est très simple. Selon la probabilité de croisement définie, chaque paramètre a une probabilité d'être croisé de $\frac{1}{2}$. La figure 5.6 montre un exemple de croisement uniforme.

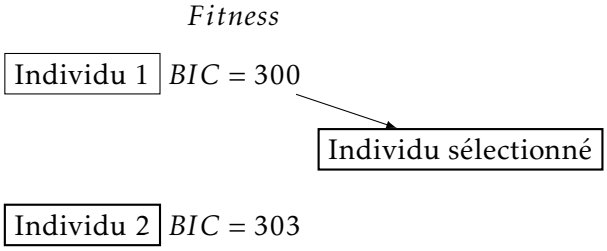


FIGURE 5.5 – Exemple de sélection par tournoi.

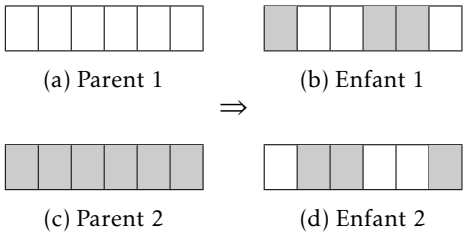


FIGURE 5.6 – Croisement uniforme.

Croisement SBX (Croisement binaire simulé) [1] [106] : Le croisement SBX repose sur la simulation du fonctionnement du croisement un point utilisé dans le cadre binaire. Le principe du croisement un point est de déterminer un point de coupure aléatoirement puis d'échanger la partie suivant ce point de coupure pour les deux parents. La figure 5.7 présente un exemple de croisement un point.

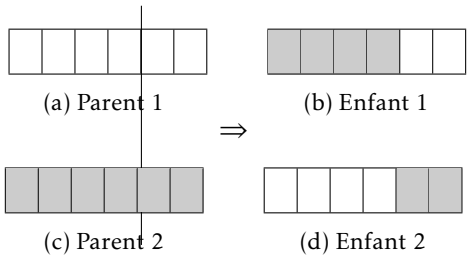


FIGURE 5.7 – Croisement un point.

Le croisement SBX est un opérateur s'adaptant à l'évolution de l'algorithme en fonction de la fonction objective des parents et des enfants. Le principe de ce

croisement est la génération de deux enfants $c_1(i)$ et $c_2(i)$ par la relation suivante :

$$\begin{cases} c_1(i) = 0.5[(1+\beta)p_1(i) + (1-\beta)p_2(i)] \\ c_2(i) = 0.5[(1-\beta)p_1(i) + (1+\beta)p_2(i)] \end{cases}$$

où $p_1(i)$ et $p_2(i)$ sont les parents générant les enfants et où β est un facteur de répartition/diffusion donné par :

$$\beta = \begin{cases} (2u)^{\frac{1}{\eta+1}} \text{ if } u < 0.5 \\ (\frac{1}{2(1-u)})^{\frac{1}{\eta+1}} \text{ sinon.} \end{cases}$$

où u est un nombre aléatoire uniformément généré dans l'intervalle $[0,1]$ et η un paramètre réel non négatif.

Remarque : La loi de probabilité utilisée pour la génération des solutions enfants est dérivée afin d'obtenir un pouvoir de recherche similaire au croisement un point utilisé dans le cadre binaire.

Pour atteindre l'objectif d'estimation des probabilités de transition, il a été décidé d'utiliser un codage réel pour la représentation des solutions. Néanmoins, la contrainte des zéros fixée dans la section 3.1.3 afin d'avoir des modèles identifiables, requiert l'adaptation de ces opérateurs. Les opérateurs de croisement et de mutation peuvent faire évoluer des paramètres fixés à zéro chez les parents en paramètres estimés chez les enfants. Le modèle venant des parents est alors totalement modifié chez les enfants. Il est alors nécessaire d'évaluer la pertinence de l'évolution des paramètres fixés à zéro venant des parents. Pour cela, deux méthodes sont comparées.

Méthode 1 : L'opérateur de croisement est appliqué uniquement lorsque les paramètres des deux parents sont estimés.

L'exemple Figure 5.8, représente le processus de la méthode 1. Dans cet exemple, le croisement SBX a été adapté afin de n'être appliqué que lorsque le paramètre est différent de zéro. Pour les deux premières modalités (T et T++) de la variable **x**, l'enfant (tableau c), reprend les valeurs du parent 1 (tableau a) sans changement. Pour ces deux modalités, les deux seules possibilités de modification sont indiquées par les paramètres en rouge. Ce sont les seuls paramètres qui sont à la fois estimés pour le parent 1 (tableau a) et le parent 2 (tableau b). Les paramètres en gras dans le tableau (c), indiquent les paramètres ayant été croisés. Il est alors possible de constater que ces paramètres sont bien estimés à la fois pour le parent 1 et pour le parent 2. D'autre part cet, exemple permet de constater que lorsque

x\y	T	T+	T++	I	TR	TR+	TR++
T	0	0	0	0	0.9	0.1	0
T++	0.19	0.07	0.73	0	0	0	0
I	0	0.15	0	0.27	0	0	0.57
TR	0	0	0.024	0.097	0	0	0

(a) Parent 1

x\y	T	T+	T++	I	TR	TR+	TR++
T	1	0	0	0	0	0	0
T++	0	0.25	0.4	0	0	0	0.35
I	0	1	0	0	0	0	0
TR	0	0.46	0.17	0.21	0.32	0.15	0

(b) Parent 2

x\y	T	T+	T++	I	TR	TR+	TR++
T	0	0	0	0	0.9	0.1	0
T++	0.19	0.07	0.73	0	0	0	0
I	0	**0.97**	0	0.27	0	0	0.57
TR	0	0	**0.18**	**0.16**	0	0	0

(c) Enfant généré

FIGURE 5.8 – Exemple avec le croisement appliqué uniquement sur les paramètres estimés.

l'un des deux parents comporte un paramètre fixé à zéro, le paramètre n'est pas croisé chez l'enfant.

Méthode 2 : Dans cette méthode, l'opérateur de croisement est appliqué sans contrainte, c'est-à-dire qu'un paramètre fixé à zéro chez l'un des deux parents, peut devenir un paramètre estimé chez l'enfant et vice-versa.

La Figure 5.9 montre le processus de la seconde méthode. A l'instar de la première méthode, un croisement SBX est utilisé. Pour cette méthode, le croisement n'a pas été adapté et est applicable sur tous les paramètres de la solution. Les paramètres en gras dans le tableau (c) représentent les paramètres où l'opérateur de croisement a été appliqué. Avec le premier paramètre de la modalité T de la variable **x**, la différence entre les deux méthodes peut être constatée. En effet, alors que le paramètre est fixé à zéro pour le parent 2 (tableau (b) de la figure 5.9), l'opérateur de croisement a été appliqué puisque le paramètre s'est transformé en zéro chez l'enfant. Le même phénomène est remarquable pour tous les paramètres. Dans cette méthode, l'information sur la place des paramètres égaux à zéro n'est donc pas gardée.

Les deux méthodes seront comparées afin de trouver la plus pertinente concernant l'évolution des paramètres fixés à zéro. La première étant de ne pas croiser les paramètres égaux à zéro, contrairement à la seconde méthode autorisant le croisement sur tous les paramètres.

x\y	T	T+	T++	I	TR	TR+	TR++
T	0.01	0	0.97	0	0.002	0.016	0
T++	0	0	0	0.62	0.14	0	0.23
I	0	0.67	0	0.33	0	0	0
TR	0	0	0	1	0	0	0

(a) Parent 1

x\y	T	T+	T++	I	TR	TR+	TR++
T	0	0.29	0.5	0.21	0	0.02	0
T++	0	0	1	0	0	0	0
I	1	0	0	0	0	0	0
TR	0	0.002	0	0.03	0.9	0	0.09

(b) Parent 2

x\y	T	T+	T++	I	TR	TR+	TR++
T	**0**	**0.29**	0.95	0.013	0.002	0.016	0
T++	0	0	0	0.021	0.15	0	0.23
I	1	**0.01**	0	**0.016**	0	0	0
TR	0	0	0	1	**0.07**	0	**0.1**

(c) Enfant généré

FIGURE 5.9 – Exemple où le croisement est estimé sur tous les paramètres.

Opérateur de mutation

Concernant l'opérateur de mutation, la mutation dite "polynomiale" [106], [29] est choisie. Pour cet opérateur, une loi de probabilité polynomiale est utilisée afin de perturber une solution se situant dans le voisinage des parents. La distribution de probabilité est ajustée à gauche et à droite d'une valeur variable afin qu'aucune valeur à l'extérieur d'un intervalle spécifique [a, b] ne soit créée par l'opérateur de mutation. Dans la mutation polynomiale, l'enfant x_i' venant d'un parent x_i est généré comme suit :

$$x_i' = x_i + (x_i^U - x_i^L)w_i \tag{5.3}$$

où x_i^U (resp. x_i^L) représente la borne supérieure (resp. borne inférieure) pour x_i. Le paramètre w_i est calculé tel que la loi de probabilité polynomiale soit : $p(w) = 0.5(\eta_m + 1)(1-|w|^{\eta_m})$

$$w_i = \begin{cases} (2r_i)^{\frac{1}{\eta_m+1}} \text{ if } r_i < 0.5 \\ 1 - (2(1-r_i))^{\frac{1}{\eta_m+1}} \text{ sinon.} \end{cases}$$

où η_m est l'index de distribution et r_i est un nombre aléatoire dans l'intervalle [0,1].

Remarque : L'opérateur de croisement SBX et l'opérateur de mutation polynomiale favorisent les enfants proches de leurs parents.

5.4 Algorithme de correction

L'utilisation des probabilités de transition implique de travailler avec la contrainte suivante : pour chaque modalité de la variable $\boldsymbol{x}$, la somme des probabilités de transition doit être égale à 1 ($\sum_{h'=1}^{q} = 1$). Les solutions initiales respectent cette contrainte. Cependant, l'évolution réalisée par les opérateurs de croisement et de mutation implique que cette contrainte ne soit plus respectée par les enfants générés. Ces opérateurs ne prenant pas en compte cette contrainte. Les tableaux 5.8 (c) et 5.9 (c) permettent de constater le non-respect de cette contrainte. Un opérateur de correction est alors appliqué sur les enfants générés afin qu'ils respectent cette contrainte. Cet opérateur correspond à l'algorithme 3 et est résumé par la figure 5.10. Il est appliqué après les opérateurs classiques de croisement et de mutation, comme que l'indique la figure 5.11.

Fonctionnement : Pour chaque modalité de la variable $\boldsymbol{x}$, la somme des probabilités de transition est calculée. Si cette somme est égale à 1, il ne fait rien et passe à la suite. Sinon, la valeur initiale du paramètre estimé est ajustée par une pondération.

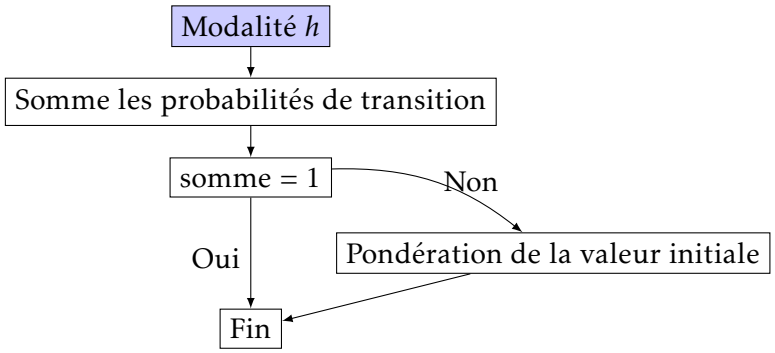


Figure 5.10 – Processus de l'opérateur de correction.

Le tableau 5.1 correspond à l'enfant représenté par la figure 5.8 (tableau (c)). Le tableau 5.2 présente son évolution suite à l'application de l'algorithme de correction. Les paramètres en gras correspondent aux paramètres où l'opérateur de correction a été appliqué. Suite à l'application de l'opérateur de correction, la somme des paramètres pour la modalité I de la variable $\boldsymbol{x}$ dans le tableau 5.2 est égal à 1 : $0.54 + 0.15 + 0.31 = 1$ alors qu'avant l'application de l'opérateur, elle était égale à 1.8 dans le tableau 5.1. Le même constat peut être réalisé pour la modalité TR de la variable $\boldsymbol{x}$.

Les différents opérateurs possibles ayant été définis et sélectionnés, 8 algorithmes génétiques peuvent être créés. L'objectif est désormais de comparer ces 8 algorithmes,

Algorithme 3 : Algorithme de correction

```
Data : entier sup, i, j;
réel différence, sum;
réel value;
for Chaque modalité de x do
    sum=0;
    for Chaque modalité de y do
        Calcul de la somme
    end
    if (sum ≥ 1) then
        sup=1;
    else
        sup=0;
    end
    différence = Math.abs(sum − 1)
end
for Chaque modalité de x do
    value = Valeur de la solution à l'indice(i+j*q);
    if (sup == 1) then
        mise à jour de la valeur par (value-((différence*value)/sum));
    else
        mise à jour de la valeur par (value+((différence*value)/sum));
    end
end
```

$X \backslash Y$	T	T+	T++	I	TR	TR+	TR++
T	0	0	0	0	0.9	0.1	0
T++	0.19	0.07	0.73	0	0	0	0
I	0	**0.97**	0	0.27	0	0	0.57
TR	0	0	**0.18**	**0.16**	0	0	0

Tableau 5.1 – Enfant généré dans l'exemple de la figure 5.8 (tableau (c)).

$X \backslash Y$	T	T+	T++	I	TR	TR+	TR++
T	0	0	0	0	0.9	0.1	0
T++	0.19	0.07	0.73	0	0	0	0
I	0	**0.54**	0	**0.15**	0	0	**0.31**
TR	0	0	**0.1**	**0.9**	0	0	0

Tableau 5.2 – Enfant généré après l'application de l'opérateur de correction.

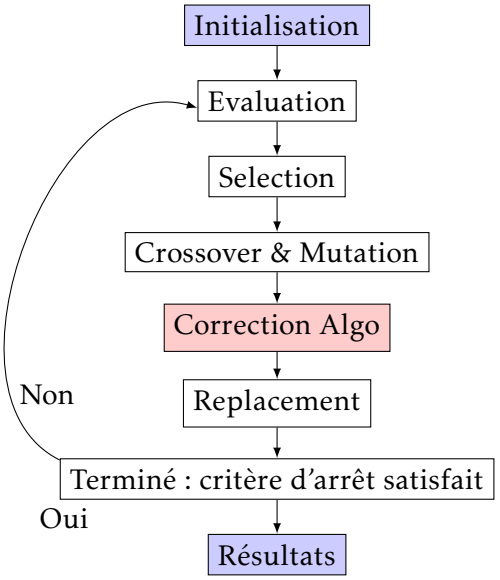


FIGURE 5.11 – Processus de l'algorithme génétique avec l'opérateur de correction.

correspondant aux différentes combinaisons d'opérateurs. La comparaison des algorithmes et des différents opérateurs est réalisée à l'aide de plusieurs expérimentations qui sont présentées dans la section suivante.

5.5 Expériences

Ces expérimentations ont pour objectif de trouver la méta-heuristique la plus efficace en fonction des différentes combinaisons des opérateurs. 8 algorithmes génétiques sont comparés.

5.5.1 Protocole expérimental

Pour comparer les 8 algorithmes génétiques, un jeu de données simulées a été généré à l'aide du logiciel R et de la fonction Rmultinorm (voir section 5.5.2). Chaque méta-heuristique est implémentée en JAVA avec la plate-forme JMETAL [34] sur une machine Linux ayant un processeur Intel Core i5-4590 CPU 3.3GHz*4 et 4 GO de RAM. Chaque algorithme génétique est lancé 25 fois sur le jeu de données simulé. Pour chacune des 25 exécutions des 8 algorithmes génétiques, la meilleure solution trouvée est sauvegardée, puis la moyenne des 25 meilleures solutions est calculée pour chaque méta-heuristique. Le critère d'arrêt pour chaque algorithme est un nombre maximum d'itérations. Afin de comparer les performances des différentes méta-heuristiques, un

test statistique de comparaison de moyenne (Kruskal Wallis) est effectué. Ce test a été choisi car l'échantillon est petit, 25 résultats, et les instances sont indépendantes. Si le test est significatif, un test unilatéral de Mann Whitney sera effectué pour avoir la méta-heuristique ayant une moyenne significativement plus petite. D'autre part, pour avoir une visualisation des résultats, un boxplot de chaque méta-heuristique est également réalisé. La méta-heuristique ayant la plus petite moyenne et étant la plus robuste sera sélectionnée pour être comparée avec les résultats de la stratégie EXBIC.

5.5.2 Description des données

Les expériences sont effectuées sur un jeu de données simulé (DS3), généré avec le logiciel R et la fonction Rmultinorm [1]. Cette fonction permet de générer des données selon une loi multinomiale. Le tableau 5.3(a) montre la représentation du jeu de données. Dans cet exemple, 10 000 couples (**x**,**y**) ont été générés. Étant donné que le modèle est simulé, les probabilités de transition et la place des paramètres fixés à zéro sont connus. Le tableau 5.3(b) montre les probabilités de transition associées au tableau 5.3(a).

$x\backslash y$	1	2	3	4	5	6	7
1	2122	1260	0	0	0	0	0
2	0	0	961	0	0	0	0
3	0	0	0	1975	0	0	0
4	0	0	0	0	3529	143	10

(a) Matrice du jeu de données simulé.

$x\backslash y$	1	2	3	4	5	6	7
1	0.63	0.37	0	0	0	0	0
2	0	0	1	0	0	0	0
3	0	0	0	1	0	0	0
4	0	0	0	0	0.96	0.038	0.002

(b) Matrice de probabilités des données simulées.

Tableau 5.3 – Jeu de données DS3.

Description du jeu de données Le tableau 5.4 décrit le jeu de donnée qui a été simulé.

5.5.3 Paramètres

En ce qui concerne les paramètres, différentes valeurs ont été étudiées avant de choisir lesquelles seraient utilisées pour les expériences finales. Tous les paramètres ont été choisis expérimentalement. Pour le nombre maximum d'itérations, une étude de

1. http ://stat.ethz.ch/R-manual/R-devel/library/stats/html/Multinom.html

Caractéristiques	Valeur
La taille de la variable $\boldsymbol{x}$	10 000
La taille de la variable $\boldsymbol{y}$	10 000
Nombre de modalité de $\boldsymbol{x}$ ($\#X_i$)	4
Nombre de modalité de $\boldsymbol{y}$ ($\#Y_i$)	7
Nombre de paramètres à estimer ($\#\mathbf{p}_.$)	6
Nombre de modèle comparés avec la méthode exhaustive ($\#\boldsymbol{\Delta}$)	4 095

Tableau 5.4 – Paramètres des algorithmes génétiques.

la convergence de l'algorithme a été effectuée. Le tableau 5.5 indique les paramètres utilisés dans cette étude.

Paramètres	Valeur
Nombre maximum d'itérations	100 000
Taille de la population	15 000
Probabilité de croisement	0.8
Probabilité de mutation	0.8
Index de distribution de croisement	20
Index de distribution de mutation	20

Tableau 5.5 – Paramètre de l'algorithme génétique.

5.5.4 Analyse de sensibilité des opérateurs

Les 8 méta-heuristiques comparées et leurs caractéristiques sont décrites dans le tableau 5.6 :

Nom	Méthode	Croisement	Mutation
C1Ua	1	Uniforme	Plusieurs paramètres
C1U	1	Uniforme	Un seul paramètre
C2Ua	2	Uniforme	Plusieurs paramètres
C2U	2	Uniforme	Un seul paramètre
CSBX1a	1	SBX	Plusieurs paramètres
CSBX1	1	SBX	Un seul paramètre
CSBX_class_a	2	SBX	Plusieurs paramètres
CSBX_class	2	SBX	Un seul paramètre

Tableau 5.6 – Composants des algorithmes génétiques.

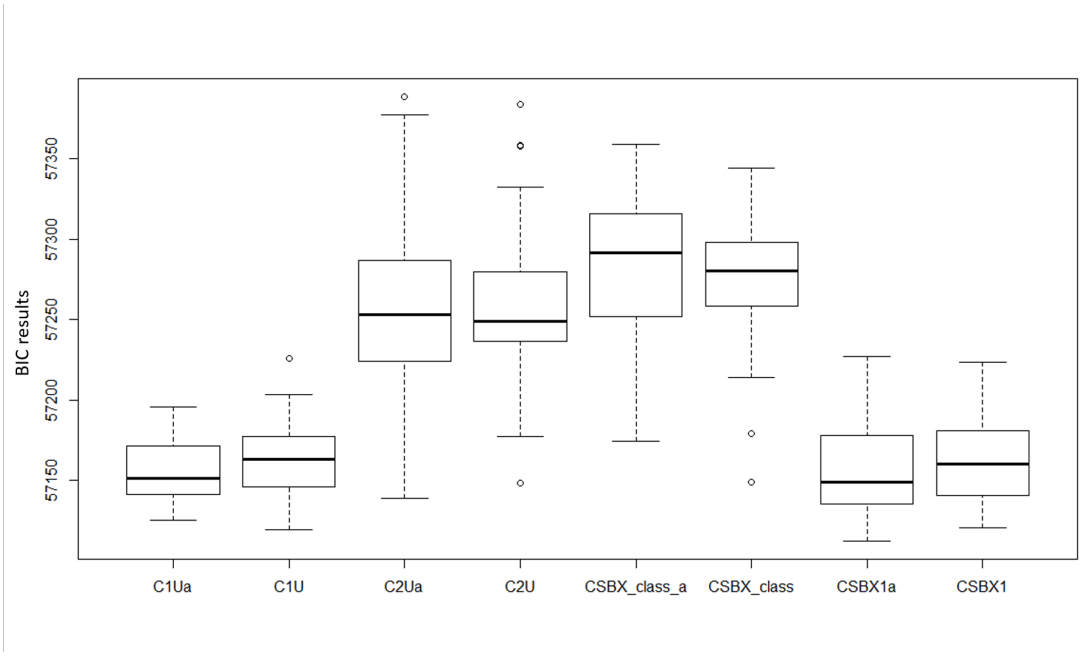


FIGURE 5.12 – Boxplot des 8 méta-heuristiques comparées.

La méthode 1 correspond à la méthode où le croisement est possible uniquement sur les paramètres estimés. La méthode 2 correspond à la méthode où le croisement est possible sur tous les paramètres de la solution.

Le boxplot Figure 5.12 présente les valeurs du critère BIC pour les 8 méta-heuristiques comparées. Les méta-heuristiques C2Ua, C2U, CSBX_class et CSBX_class_a correspondent aux méta-heuristiques qui utilisent la méthode 2, soit la proposition où le croisement est permis sur tous les paramètres (fixés à zéro et estimés). Avec ce boxplot, leurs résultats semblent moins bons que les résultats des autres méta-heuristiques. En effet, les valeurs du critère BIC trouvées semblent plus élevées. Hors, l'objectif est de minimiser ce critère. Les méta-heuristiques qui semblent avoir de meilleurs résultats correspondent aux méta-heuristiques où l'opérateur de croisement est appliqué seulement sur les paramètres estimés. La première idée serait d'utiliser cet opérateur de croisement. Les méta-heuristiques avec une mutation polynomiale appliquée sur plusieurs paramètres (C1Ua, CSBX1a) semblent avoir des résultats légèrement meilleurs que les résultats des méta-heuristiques avec une mutation polynomiale appliquée sur un seul paramètre.

Plusieurs tests statistiques ont été réalisés pour comparer ces méta-heuristiques et savoir si les différences sont significatives. Le premier test appliqué est un test de

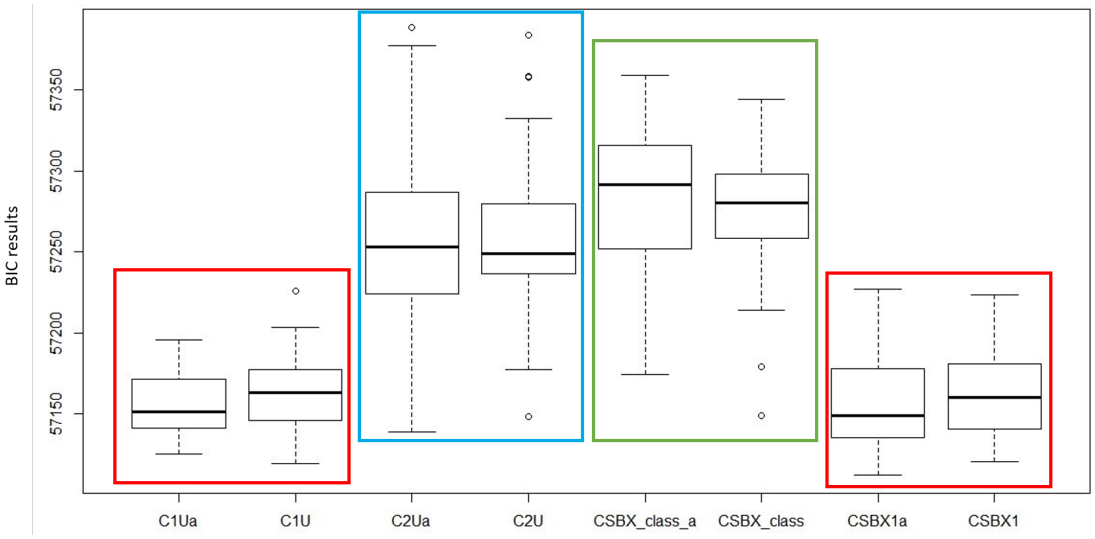


FIGURE 5.13 – Boxplot des 8 méta-heuristiques comparées.

Kruskal Wallis. Ce test permet de comparer l'ensemble des 8 méta-heuristiques et de savoir s'il y a réellement une différence significative entre les résultats des méta-heuristiques. Pour ce test, les hypothèses sont les suivantes :

$$\begin{cases} H0 : \text{Tous les résultats des méta-heuristiques sont similaires} \\ H1 : \text{Il y a des différences entre les résultats des méta-heuristiques.} \end{cases}$$

Le résultat de ce test est $p.value < 2.2e-16$. Ce résultat de $p.value$ indique que l'hypothèse nulle est rejetée par le test. Il y a réellement une différence significative entre les résultats des méta-heuristiques. Le test de **Kruskal Wallis** indiquant une différence significative entre les résultats des méta-heuristiques, ce résultat est à affiner. Pour cela, un test de **Mann Withney** est effectué. Pour ce test, les méta-heuristiques sont comparées deux par deux. Dans un premier temps, un test bilatéral a été réalisé. Ses hypothèses sont les suivantes :

$$\begin{cases} H0 : \text{Le résultat de la méta-heuristique (1) = Le résultat de la méta-heuristique (2)} \\ H1 : \text{Le résultat de la méta-heuristique (1)} \neq \text{Le résultat de la méta-heuristique (2).} \end{cases}$$

Le test indique que les méta-heuristiques peuvent être groupées en 3 sous-groupes, qui sont : {C1Ua, C1U, CSBX1a, CSBX1},{C2Ua, C2U} et {CSBX_class_a, CSBX_class}. Les différents groupes sont indiqués sur la figure 5.13. Le premier groupe, en rouge, regroupe les méta-heuristiques ayant les résultats les plus bas. C'est à dire, les méta-

heuristiques trouvant les plus faibles valeurs pour le critère BIC. Notre objectif étant de minimiser la valeur du critère BIC, ce groupe est particulièrement intéressant. Parmi ces 4 méta-heuristiques, un dernier test est réalisé afin de détecter si l'une de ces 4 méta-heuristiques donne des résultats significativement inférieurs aux autres. Ce test est un test unilatéral gauche de Mann Withney. Les résultats de ce test indiquent, qu'il n'y a pas de différence significative entre les 4 méta-heuristiques. Cela signifie que pour la suite, 4 méta-heuristiques ayant des performances similaires peuvent être comparées à la stratégie EXBIC. Il est à noter que les 4 méta-heuristiques donnant les meilleures performances sont celles où la méthode 1 a été utilisée. C'est à dire la méthode où l'opérateur de croisement a été appliqué uniquement sur les paramètres estimés. Cela confirme que l'information sur la position des paramètres fixés à zéro est importante et pertinente dans notre algorithme génétique. Au contraire, le fait d'appliquer l'opérateur de mutation sur un ou plusieurs paramètres ne semble pas changer significativement les résultats.

5.6 Conclusion

Dans le chapitre 3 et 4, nous avons proposé la méthode **EXsearch**, et les stratégies EXBIC et EXBIL associées, pour estimer et comparer un certain nombre de modèles. Selon la variable étudiée, le problème combinatoire peut rapidement devenir conséquent et la méthode **EXsearch** très consommatrice en temps. Dans ce chapitre, nous proposons d'utiliser une méthode d'optimisation stochastique pour surpasser les défauts de la méthode **EXsearch**. C'est à dire, avoir une méthode plus rapide et pouvant comparer un plus grand nombre de modèles. Pour la méthode non-exhaustive, deux problématiques ont émergé. La première étant que nous voulons résoudre dans un même algorithme un problème à la fois continu et combinatoire. En effet, l'estimation de paramètres correspond à un problème continu alors que la sélection de modèles correspond à problème combinatoire. La seconde problématique est que pour chaque modèle les paramètres doivent être estimés, donc il y a autant de problèmes continus que de modèles à estimer. Pour résoudre ces problématiques, nous avons choisi d'utiliser un algorithme génétique d'état stationnaire. Dans l'algorithme génétique proposé, une solution correspond à un modèle composé des probabilités de transition estimées et des paramètres fixés à zéro. Pour garder l'information des paramètres fixés à zéro et donc leur position, un nouvel opérateur de croisement a été proposé. Cet opérateur est comparé à un opérateur de croisement classique appliqué sur tous les paramètres. Finalement, pour trouver et garder l'algorithme le plus efficace, 8 méta-heuristiques ont été comparées.

Comme chaque solution correspond à un modèle probabiliste, un nouvel opérateur de correction a été créé et est appliqué après chaque opérateur de croisement et de mutation. Cet opérateur de correction permet de garder la somme des probabilités pour chaque modalité de **x** égales à 1. Le résultat de la comparaison montre que l'algorithme génétique ayant l'opérateur de croisement appliqué uniquement sur les probabilités de transition estimées est plus efficace que l'algorithme génétique ayant l'opérateur de croisement appliqué sur tous les paramètres. Le principal objectif de ce chapitre était de proposer une méthode pour challenger les résultats de la méthode **EXsearch**. 8 méta-heuristiques ont été comparées et nous avons montré que 4 d'entre elles avaient de meilleures performances que les autres. Pour le chapitre suivant, nous sélectionnerons une méta-heuristique parmi les 4 méta-heuristiques les plus performantes afin de comparer la méthode **AGsearch** à la méthode **EXsearch**. Le chapitre suivant définit et compare les différentes stratégies possibles reprenant les méthodes et critères proposés dans ce travail.

Chapitre 6

Expériences numériques sur données simulées et réelles

Dans les chapitres précédents, deux méthodes d'estimation et de comparaison de modèles ont été proposées : une méthode exhaustive **EXsearch**, proposée dans le chapitre 3, et une méthode non exhaustive **AGsearch**, proposée dans le chapitre 5. Ces deux méthodes requièrent l'utilisation d'un critère de sélection de modèles. Dans le chapitre 4, deux critères ont été proposés : un critère asymptotique **BIC** et un critère non asymptotique **BIL**. Dans ce dernier chapitre, les performances des différentes stratégies possibles, sont comparées, le but étant de sélectionner la stratégie ayant le meilleur compromis "rapidité-résultats". Ces stratégies sont évaluées sur quatre critères : le temps de calcul, la valeur du critère de sélection retournée, l'interprétabilité du modèle et sa capacité d'amélioration des performances dans un modèle de classification.

6.1 Stratégie

La première méthode proposée dans ce travail, **EXsearch**, est une méthode exhaustive où l'estimation et la comparaison sont réalisées sur l'ensemble des modèles de l'espace de recherche. La seconde méthode, **AGsearch**, est une méthode non exhaustive, reposant sur un algorithme génétique. Deux critères de sélection de modèles ont également été proposés : le critère BIC et le critère BIL. Quatre stratégies peuvent alors être comparées, reposant sur les différentes combinaisons possibles des méthodes de recherche et des critères de sélection de modèles.

Stratégie 1 : EXBIC La stratégie EXBIC correspond à l'utilisation de la méthode

EXsearch utilisant pour critère de sélection, le critère BIC.

Stratégie 2 : EXBIL Le critère BIL permet d'affiner le classement des modèles δ. Des modèles éloignés par le critère BIC, le seront également par le critère BIL. Le calcul du critère BIL étant très coûteux en temps, celui-ci ne sera effectué que sur les 10 modèles minimisant le plus le critère BIC. La stratégie EXBIL est alors réalisée en deux étapes. La première étape reprend la stratégie **EXBIC**. C'est à dire, l'utilisation de la méthode exhaustive et du critère BIC sur l'ensemble des modèles. Dans la seconde étape, le critère BIL est calculé pour les 10 meilleurs modèles sélectionnés par le critère BIC afin d'affiner la recherche et de sélectionner le meilleur modèle au sens du critère BIL.

Stratégie 3 : AGBIC La stratégie AGBIC correspond à l'utilisation de la méthode **AGsearch** utilisant pour critère de sélection, le critère BIC.

Stratégie 4 : AGBIL A l'instar de la seconde stratégie, la stratégie AGBIL est réalisée en deux étapes. La première étape reprend la stratégie **AGBIC**. Soit, l'utilisation de la méthode non-exhaustive avec le critère BIC comme critère de sélection. Pour comparer les modèles retournés par la stratégie AGBIC, celle-ci est lancée 10 fois. Pour chaque exécution le modèle minimisant le critère BIC est sauvegardé. Le critère BIL est ensuite calculé pour ces 10 modèles afin de sélectionner le meilleur modèle au sens du critère BIL.

6.2 Méthodologie

Dans le but de trouver la stratégie la plus rapide et donnant les meilleurs résultats, les performances des différentes stratégies sont comparées. Dans un premier temps, les performances des critères de sélection sont comparées. L'utilisation d'un échantillonnage préférentiel, comportant un échantillonneur Gibbs pour le calcul du critère BIL rend son temps d'exécution très coûteux. A l'inverse, le temps de calcul du critère BIC est quasi-instantané suite à l'estimation des paramètres du modèle. Le critère BIL est construit de façon à sélectionner le "quasi-vrai" modèle de l'ensemble de modèles $\mathbf{\Delta}$ quel que soit le jeu de données utilisé. A contrario, l'utilisation du critère BIC peut être limitée selon le jeu de données utilisé. L'objectif de cette première comparaison est, dans un premier temps, de vérifier les performances du critère BIL et d'évaluer les performances du critère BIC sur divers jeux de données simulés. Dans un second temps, de constater dans quelle mesure le critère BIC peut remplacer l'utilisation du critère BIL, afin d'optimiser le temps de calcul de la méthode. Suite à cette comparaison, les valeurs

de critère de sélection des méthodes **EXsearch** et **AGsearch** sont comparées, l'objectif étant d'évaluer les performance de la méthode **AGsearch**. La méthode **EXsearch** étant exhaustive, utilisée sur des données simulées comportant le "quasi-vrai" modèle, la valeur minimisant le critère de sélection correspond à la valeur du "quasi-vrai" modèle. Cette valeur sert donc de référence et l'objectif de la méthode **AGsearch** est de trouver des modèles ayant des valeurs de critères de sélection similaires à la valeur de référence. Dans un troisième, les temps d'exécutions des méthodes sont également comparés à travers les stratégies EXBIC et AGBIC. Les stratégies EXBIL et AGBIL étant exécutées sur un nombre de modèles similaires, avec les mêmes paramètres, les temps d'exécution seront similaires. A contrario, la stratégie EXBIC dépendant principalement de l'algorithme EM et la stratégie AGBIC dépendant principalement de l'algorithme génétique, il parait plus pertinent de comparer ces deux stratégies. Enfin, la qualité de l'estimation des paramètres est également évaluée.

6.3 Protocole expérimental

Pour effectuer les comparaisons, la méthode **AGsearch** est exécutée avec la méta-heuristique CSBX1, présentée dans le chapitre 5. Cette méta-heuristique est implémentée en JAVA avec la plateforme JMETAL [34], sur une machine Linux ayant un processeur Intel Core i5-4590 CPU 3.3GHz*4 et 4GO de RAM. Le méthode **EXsearch** est implémentée avec le logiciel Rstudio et est exécutée sur la même machine Linux que la méthode **AGsearch**. Pour effectuer les différentes comparaisons, 16 jeux de données simulés, dont les caractéristiques sont présentées en section 6.3 ont été générés. De plus, 4 jeux de données réelles provenant de la société MeilleureAssurance.com sont également utilisés. Les caractéristiques des quatre jeux de données réelles sont également présentées dans le tableau 6.1.

Comparaison des critères de sélection : Pour effectuer la comparaison des deux critères de sélection de modèles (BIC et BIL) les 16 jeux de données simulés sont utilisés. La comparaison des performances des deux critères est effectuée avec la méthode **EXsearch** et les stratégies **EXBIC** et **EXBIL**. Ces stratégies ont l'avantage de comparer l'ensemble des modèles proposés parmi lesquels se trouvent le modèle ayant servi à générer les données simulées. Pour chacun des 16 jeux de données simulés, les stratégies **EXBIC** et **EXBIL** ont été exécutées une fois. Le modèle sélectionné par chacun des deux critères est ensuite comparé au modèle ayant servi à générer les données. Les performances des deux critères sont alors

évaluer sur le nombre "vrai" modèles retrouvés parmi les 16 jeux de données simulés.

Comparaison des méthodes EXsearch et AGsearch : L'objectif de cette comparaison étant d'évaluer les performances de la méthode non-exhaustive, les 16 jeux de données simulés sont de nouveau utilisés. Pour cette comparaison, quatre jeux de données réelles sont également utilisés. Pour chaque jeu de données, la méthode **AGsearch** est exécutée 10 fois et est arrêtée après un nombre maximum d'itérations. Pour chacune des exécutions, la meilleure solution est sauvegardée. La valeur minimisant le critère de sélection parmi les 10 valeurs sauvegardées est comparée à la valeur du critère de sélection retournées par la méthode **EXsearch**, pour chacun des jeux de données. Un test de **Mann Whitney**, est ensuite effectué pour évaluer la similarité des résultats.

Comparaison des temps de calculs : De même que pour les comparaisons précédentes, les 16 jeux de données simulées et les quatre jeux de données réelles. Cette comparaison ayant pour but de comparer la rapidité des méthodes proposées, pour chaque jeu de données, les temps de calculs des stratégies **EXBIC** et **AGBIC** sont calculés. Le ratio entre les deux méthodes utilisant le critère BIC est également calculé pour chacun des jeux de données.

Estimation des paramètres : Les valeurs des paramètres estimés par l'algorithme EM et l'algorithme génétique sont également comparées sur le jeu de données simulé "DS_35_7", afin d'évaluer la qualité de l'estimation effectuée par les méthodes **EXsearch** et **AGsearch**.

6.4 Données

Les expérimentations sont effectuées sur 16 jeux de données simulés et 4 jeux de données réels, provenant de la société MeilleureAssurance.com. Les jeux de données simulés sont générés avec le logiciel R et la fonction Rmultinorm [1]. (Voir section 5.5.2). Le tableau 6.1 détaille les caractéristiques des différents jeux de données utilisés pour les expériences. Pour chaque jeu de données, le tableau 6.1 indique : la taille de l'échantillon n (i.e., le nombre d'observations), le nombre de modalités (p) et (q), le nombre de paramètres à estimer (#$\mathbf{p}$), le nombre de modèles comparés par la méthode **EXsearch** (#$\mathbf{\Delta}$) et si le jeu de données simulé contient des paramètres situés sur les bords de l'espace des paramètres (Esp).

1. http ://stat.ethz.ch/R-manual/R-devel/library/stats/html/Multinom.html

Name	n	p	q	#p.	#Δ	Esp
Jeux de données simulés						
DS_47_1	15000	4	7	6	4095	non
DS_47_2	10035	4	7	6	4095	oui
DS_47_3	10000	4	7	6	4095	oui
DS_47_4	5000	4	7	6	4095	non
DS_35_5	15000	3	5	4	81	non
DS_35_6	10000	3	5	4	81	non
DS_35_7	5000	3	5	4	81	non
DS_34_8	15000	3	4	3	26	non
DS_34_9	7880	3	4	3	26	oui
DS_34_10	5000	3	4	3	26	non
DS_76_11	5000	7	6	6	16807	oui
DS_64_12	5000	6	4	3	218	non
DS_64_13	10000	6	4	3	218	oui
DS_35_14	500	3	5	4	81	non
DS_34_15	1000	3	4	3	26	non
DS_35_16	100	3	5	3	81	non
Jeux de données réelles						
NGS	11441	4	7	6	4095	-
CS	8238	3	5	4	81	-
NGS2	7776	7	6	6	16807	-
NGS3	12437	6	4	3	218	-

Tableau 6.1 – **Description des instances**

6.5 Paramètres

Les algorithmes utilisés pour les méthodes **EXsearch** et **AGsearch** nécessitent de déterminer différentes valeurs de paramètres, indiquées dans les tableau 6.2 et 5.4.

Paramètres de la méthode EXsearch : L'algorithme EM, utilisé dans la méthode **EXsearch** nécessite de fixer le nombre maximum d'itération qu'il doit effectuer. Dans le chapitre 3, il a été montré que la vitesse de convergence de ce critère dépend de la taille de l'échantillon. Les nombre d'itération de l'algorithme a été défini empiriquement en fonction de la taille du jeu de donné. Dans la section 4.5, il a été montré que l'échantillonneur de Gibbs atteignait la stationnarité rapidement, le temps de chauffe et le nombre d'itérations de l'algorithme ont alors été fixés en conséquence. Dans la section 4.5, il a été montré que la convergence de l'estimateur BIL ne dépendait ni de la taille de l'échantillon, ni du nombre de

modalités du modèle. Différentes valeurs de paramètres ont alors été étudiées avant de décider qu'elle serait la valeur du nombre d'itérations nécessaire à sa convergence, utilisée pour les expériences finales.

Paramètres de la méthode EXsearch	**Valeur**
Nombre maximum d'itérations pour EM où $n \geq 5000$	10 000
Nombre maximum d'itérations pour EM où $n \leq 5000$	30 000
Nombre d'itérations Gibbs	500
Temps de chauffe du Gibbs	10 itérations
Nombre d'intérations Importance sampling	5 000

Tableau 6.2 – Paramètre requis pour la méthode exhaustive

Paramètres de la méthode AG : De même que pour les stratégies EXBIC et EXBIL, l'algorithme génétique requière de fixer les valeurs de certains paramètres. Différentes valeurs de paramètres ont été étudiées avant de décider lesquels seraient utilisés pour les expériences finales. Le nombre maximum d'itérations a été fixé selon une étude de la convergence de l'algorithme. Le nombre maximum d'itérations et la taille de la population diffère selon la taille de l'espace de recherche. Le tableau 6.3 indique les paramètres impliqués dans cette étude.

Paramètres de la méthode AGsearch	**Valeur**
Nombre maximum d'itérations pour les jeux de données avec $p \geq 4$ et $q \geq 6$.	100 000
Nombre maximum d'itérations pour les autres de données	50 000
Taille de la population pour les jeux de données avec $p \geq 4$ et $q \geq 6$	15 000
Taille de la population pour les autres jeux de données	10000
Probabilité de croisement	0.8
Probabilité de mutation	0.8
Index de distribution de croisement	20
Index de distribution de mutation	20

Tableau 6.3 – Paramètre des algorithmes génétiques

6.6 Comparaison

6.6.1 Comparaison des performances des critères BIC et BIL

Dans un premier temps, nous comparons les performances des critères BIC et BIL. Dans la section 4.5.5, il a été montré que le critère BIL est plus efficace que le critère

BIC sur des échantillons de petites tailles. Cependant, l'utilisation d'un échantillonnage préférentiel, comportant un échantillonneur de Gibbs implique un temps de calcul coûteux dû au nombre d'itérations requis par les deux algorithmes. D'autre part, le calcul du critère BIC est plus rapide, néanmoins, sa capacité à retrouver le vrai le modèle lorsque les paramètres sont sur le bord de l'espace n'est pas garantie. Parmi les jeux de données, certains sont de petites tailles ($n \in [100, 500]$) et d'autres ont des paramètres estimés sur le bord de leurs espaces, indiqués dans le tableau 6.1 par la colonne "Esp". Le tableau 6.4, nous permet de comparer la capacité du critère BIC à retrouver le "vrai" modèle selon les différents jeux de données simulés.

Dataset	EXBIC	EXBIL
DS_47_1	oui	oui
DS_47_2	oui	oui
DS_47_3	oui	oui
DS_47_4	oui	oui
DS_35_5	oui	oui
DS_35_6	oui	oui
DS_35_7	oui	oui
DS_34_8	oui	oui
DS_34_9	oui	oui
DS_34_10	oui	oui
DS_76_11	oui	oui
DS_64_12	oui	oui
DS_64_13	oui	oui
DS_35_14	oui	oui
DS_34_15	oui	oui
DS_35_16	non	oui

Tableau 6.4 – Résultats de la comparaison de critères BIC et BIL sur la méthode EX-search.

Le tableau 6.4 permet de constater que le critère BIL retrouve le modèle simulé pour tous les jeux de données. Le critère BIC permet également de retrouver le modèle simulé dans la majorité des cas. Le seul jeu de données où le critère BIC n'a pas sélectionné le modèle simulé est le jeu de donnée ayant une taille d'échantillon très petite ($n = 100$), indiqué en rouge. Ce qui confirme les expériences réalisées en section 4.5.5. D'autre part, dans la section 4.5.5, il avait également été indiqué que malgré l'absence de garanties théoriques, le critère BIC sélectionnait le modèle simulé pour des jeux de données ayant des paramètres sur le bord de leur espace. Le tableau 6.4 permet de réaliser un constat

similaire. Le tableau 6.1 indique que les jeux de données : DS_47_2, DS_47_3, DS_34_9, DS_76_11 et DS_64_13 ont été simulés en ayant des paramètres sur le bord de leur espace. Pour chacun de ces jeux de données, le critère BIC a sélectionné le modèle simulé.

Ces expériences montrent que le critère BIC peut être utilisé sur la plupart des échantillons de données, cependant sur les échantillons de taille $n < 500$ il vaut mieux utiliser le critère BIL, bien que plus coûteux en tant de calcul. Le choix du critère, et donc la stratégie utilisée, va alors dépendre de la taille de l'échantillon disponible. Le critère BIC montrant de bonnes performances sur la plupart des jeux de données, il sera utilisé pour la comparaison des méthodes **EXsearch** et **AGsearch** en terme de valeur de critère de sélection.

6.6.2 Comparaison des méthodes EXsearch et AGsearch

Les méthodes **AGsearch** et **EXsearch** sont comparées selon le critère de sélection BIC. Le critère BIC, ne sélectionnant pas le modèle simulé pour le jeu de données "DS_34_15", celui-ci n'apparaît pas pour la comparaison. Pour l'ensemble des autres jeux de données, le critère sélectionne le modèle simulé, sa valeur est alors utilisée en tant que valeur de référence pour la comparaison. Le critère de sélection utilisé étant le critère BIC, les stratégies EXBIC et AGBIC sont comparées. Le tableau 6.5 présente les valeurs minimisant la valeur du critère BIC pour chacune des deux stratégies.

Dans le tableau 6.5, les valeurs en gras indiquent la valeur minimale du critère BIC, quel que soit la stratégie utilisée. Il est alors possible de constater que, parmi les différents jeux de données simulés, la stratégie AGBIC trouve des résultats très proches, mêmes similaires aux valeurs du critère BIC de la stratégie EXBIC. Un test de Mann Whitney a alors été effectué afin de valider qu'il n'y a pas de différences significatives entre les résultats. Avec une $p-value = 0.9396$, l'hypothèse H0 du test est acceptée. Cela signifie qu'il n'y a pas de différence significative entre les résultats des deux stratégies. Pour certains jeux de données simulés, la valeur du critère BIC de la stratégie AGBIC est même légèrement inférieure à la valeur du critère BIC de la méthode EXBIC. Cela s'explique par la précision de la valeur des paramètres estimée par les deux algorithmes. Pour les jeux de données réels : NGS1, NGS2 et NGS3, les valeurs du critère BIC retournées par la stratégie AGBIC sont également inférieures à la valeur du critère BIC de la méthode EXBIC, avec une différence plus grande que les jeux de données simulés. La stratégie AGBIC a potentiellement trouvé un modèle meilleur que celui sélectionné par la stratégie EXBIC. Cela s'explique par le fait que l'algorithme génétique compare plus

Dataset	EXBIC	AGBIC
DS_47_1	**87 008.41**	87 018.41
DS_47_2	**62 637.1**	62 640.8
DS_47_3	**57 101.31**	57 105.8
DS_47_4	**30 770.08**	**30 766.48**
DS_35_5	**64 084.99**	**64 084.99**
DS_35_6	**49 875.41**	**49 875.41**
DS_35_7	**23 605.79**	**23 604.8**
DS_34_8	**67 325.20**	**67 325.2**
DS_34_9	**32 836.38**	**32 836.38**
DS_34_10	**21 358.85**	**21 358.85**
DS_76_11	**35 423.32**	**35 419.54**
DS_64_12	28 660.87	**28 597.87**
DS_64_13	**55 948.23**	**55 948.23**
DS_35_14	**2 436.97**	**2 436.97**
DS_34_15	4 497.59	4 497.59
NGS1	58 306.6	**58 287.7**
CS	25 548.3	25 558.9
NGS2	53 072.45	**52 895.5**
NGS3	76 387.38	76 102.8

Tableau 6.5 – Résultats des critères BIC pour les deux méthodes.

de modèles, potentiellement meilleurs que ceux disponibles pour la stratégie EXBIC.

La stratégie AGBIC, bien que reposant sur une méthode non-exhaustive donne des valeurs de critère BIC similaires aux valeurs de critère BIC de la stratégie EXBIC, pour les jeux de données simulées et même des valeurs inférieures sur les jeux de données réels. La stratégie AGBIC est donc aussi performante que la stratégie EXBIC sur des jeux de données simulées, et potentiellement meilleure sur des jeux de données réels. D'autre part, il est possible que l'algorithme génétique trouve des modèles ayant des valeurs de critères BIC très proches de la valeur du "vrai" modèle, sélectionné par la stratégie EXBIC, mais que le vrai modèle ne soit pas dans la liste de modèle renvoyé par la stratégie AGBIC. Cela peut également se produire dans le cas de modèles non identifiables. Ces deux stratégies ont donc leurs propres avantages et inconvénients, et ont des performances similaires en termes de valeurs de critères de sélection. L'objectif étant de trouver la stratégie avec le meilleur compromis "résultats-rapidité", le temps d'exécution des deux stratégies est désormais comparé.

6.6.3 Comparaison des temps de calcul des deux méthodes

Le temps d'exécution des deux stratégies est présenté dans le tableau 6.6. La colonne "ratio" indique le ratio entre les temps d'exécutions des deux stratégies.

Dataset	Méthode **EXsearch**	Méthode **AGsearch**	ratio
DS_47_1	7h	2.10min	**200**
DS_47_2	7h	2.74 min	**153.3**
DS_47_3	7h	2.5min	**168**
DS_47_4	21h	2.47 min	**565**
DS_35_5	7min	36sec	11.67
DS_35_6	7min	37sec	11.05
DS_35_7	20min	38sec	31.58
DS_34_8	2.30min	31sec	4.5
DS_34_9	2.30min	30sec	4.6
DS_34_10	6min	28sec	12.86
DS_76_11	32h	2.22min	**864.9**
DS_64_12	38min	1.55min	24.5
DS_64_13	1h05	1.50min	43.3
DS_35_14	20min	38 sec	31.6
DS_35_16	20min	36sec	33.33
NGS1	7h	2.10min	**200**
CS	7 min	37sec	11.35
NGS2	32h	2.23min	**873**
NGS3	35min	2.56min	13.67

Tableau 6.6 – Temps de calcul entre les deux méthodes.

Le tableau 6.6 permet de constater que le temps d'exécution, pour l'ensemble des jeux de données, est plus rapide pour la stratégie AGBIC que pour la stratégie EXBIC. Le temps d'exécution de l'algorithme EM dépend du nombre d'itérations nécessaire à l'algorithme pour converger et du nombre de modèle comparés. Les paramètres de la méthode **EXsearch** varient en fonction de la taille de l'échantillon. Plus la taille de l'échantillon est petite, plus l'algorithme EM a besoin d'itérations pour converger. Cela explique que le temps d'exécution pour des jeux de données de taille $n \leq 5000$ soit plus élevé que lorsque $n \geq 5000$ pour la stratégie EXBIC. Pour l'algorithme génétique, le temps d'exécution dépend des paramètres déterminer selon la taille de l'espace de recherche, indépendamment de la taille de l'échantillon et du nombre de modèle de l'ensemble. Le ratio entre les temps d'exécution des deux stratégies évolue en fonction

du nombre de modèle comparé et de la taille de l'échantillon du jeu de données. Les valeurs en gras indiquent les différences les plus importantes. On peut constater que les ratios les plus importants concernent les ensembles de modèles les plus grands ($\#\mathbf{\Delta} \in [4\,095, 16\,807]$).

La stratégie AGBIC permet donc de trouver un modèle ayant des valeurs de critères BIC similaires aux valeurs de la stratégie EXBIC beaucoup plus rapidement et peut par la suite être affiné par la stratégie AGBIL. A contrario, le temps de calcul de la stratégie EXBIC, même avec un espace de modèles limité, se retrouve vite coûteux en temps de calcul pour estimer l'ensemble des modèles. La meilleure stratégie selon le compromis "rapidité-résultats" est donc la stratégie AGBIC. Une dernière comparaison est alors effectuée concernant la précision des paramètres estimés par les deux algorithmes.

6.6.4 Estimation

La Figure 6.1 présente les résultats de l'estimation des paramètres pour le jeu de données DS_35_7, pour les stratégies EXBIC et AGBIC. Le premier constat est que les deux méthodes estiment des valeurs de paramètres très proches des paramètres simulés. Néanmoins, avec la contrainte pour réduire l'espace de modèles $\mathbf{\Delta}$ dans la méthode **EXsearch**, l'algorithme EM est contraint d'estimer uniquement certains paramètres du modèle. L'algorithme génétique peut couvrir l'ensemble de l'espace de modèles $\mathbf{\Delta}$ et a une plus grande liberté sur le choix des paramètres à estimer. Selon le modèle, une estimation plus précise des paramètres peut être effectuée par la méthode **AGsearch**.

$X \backslash Y$	1	2	3	4	5
1	0.1	0	0	0	0.9
2	0	0.3	0.7	0	0
3	0.2	0	0	0.8	0

(a) Jeu de données simulé

$X \backslash Y$	1	2	3	4	5
1	0	0	0	0	1
2	0	0.3	0.7	0	0
3	0.22	0	0	0.78	0

(b) Résultat de la stratégie EXBIC

$X \backslash Y$	1	2	3	4	5
1	0.11	0	0	0	0.89
2	0	0.29	0.71	0	0
3	0.199	0	0	0.801	0

(c) Résultat de la stratégie AGBIC

FIGURE 6.1 – Résultats pour l'estimation de paramètres du jeu de données DS_35_7.

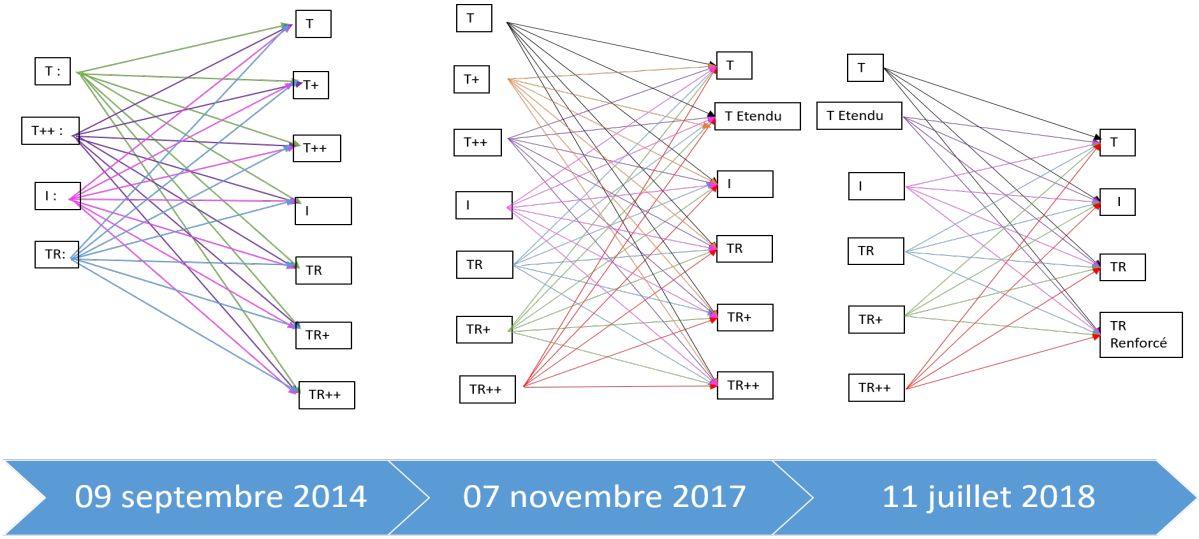


FIGURE 6.2 – Évolution variable Niveau de garantie souhaité

La stratégie AGBIC semble la plus performante, aux travers des différents critères étudiés. Cependant, l'objectif initial de ce travail est de répondre aux différents objectifs de la société MeilleureAssurance.com. Ces objectifs étant la réalisation d'un modèle prédictif robuste lors de modification d'une variable et l'interprétabilité des résultats. Les deux stratégies sont alors également comparées sur leurs performances concernant ces objectifs.

6.7 Application aux données réelles

Pour cette étude quatre jeux de données réelles, fournis par la société MeilleureAssurance.com, sont disponibles. Trois des jeux de données réelles correspondent aux trois modifications de la variable "Niveau de garantie souhaité" présentée dans la section 1.1.4 et rappelée dans cette section par la figure 1.4. Les caractéristiques de ces jeux de données sont présentées dans le tableau 6.5. Le jeu de données NGS1 correspond à la variable modifiée le 9 septembre 2014 (utilisé comme cas d'usage), le jeu de données NGS2 correspond à la variable modifiée le 7 novembre 2017 et le jeu de données NGS3 correspond à la variable modifiée le 11 juillet 2018. Le quatrième jeu de données réelles correspond à une nouvelle variable, la variable "Statut du conducteur".

Variable "Statut du conducteur" : La dernière variable étudiée concerne le statut du conducteur et la réponse à la question : "Êtes-vous actuellement assuré", toujours sur un formulaire automobile. Cette variable avait initialement trois choix

de réponses possibles ("Oui, comme conducteur principal (CP)","Oui, comme conducteur secondaire (CS)", "Non"). Suite à une modification, cinq choix de réponses sont désormais possibles : "Non, mais avant comme conducteur principal (ACP)","Non, mais avant comme conducteur secondaire (ACS)","Non", "Oui, comme conducteur secondaire (CS)","Oui, comme conducteur secondaire (CP)". La modification de cette variable est arrivée 1 mois après la première modification de la variable niveau de garantie souhaitée. Les caractéristiques de ce jeu de données, noté CS, sont présentées dans le tableau 6.5.

Ces deux variables sont toutes les deux discriminantes dans un modèle de classification. La taille des jeux de données étant assez grande ($n > 5000$), dans cette section, nous appliquons les stratégies EXBIC et AGBIC sur les quatre jeux de données réelles. Les stratégies EXBIL et AGBIL indiquant les mêmes modèles sélectionnés. Ces stratégies sont comparées sur leur interprétabilité d'un point de vue métier et sur leur capacité à améliorer un modèle prédictif. De plus, les deux méthodes retournent des modèles très proches en termes de valeur de critère BIC. Il peut alors être intéressant de ne pas utiliser l'information d'un seul modèle mais l'information de plusieurs modèles. Pour cela, un "Bayesian Model Averaging" (BMA), présenté en section 4.4 est réalisé sur les deux stratégies. Les modèles agrégés par le "Bayesian Model Averaging" de chacune des stratégies sont alors également comparés en termes d'interprétabilité et de classification.

6.7.1 Interprétabilité

L'un des objectifs de ce travail est d'avoir des modèles interprétables, d'un point de vue métier. Dans cette partie, les différents modèles retournées par les stratégies sont présentés et analysés.

NGS1 : Le tableau 6.7 indique les modèles retournés par les stratégie EXBIC, AGBIC et les Bayesian Model Averaging (BMA) associés. Trois des quatre solutions (tableau (a),(b) et (c)), indiquent que 100% des internautes ayant choisis la modalité Tous Risques (TR) avant la modification choisiraient de nouveau la modalité Tous Risques, s'ils revenaient avec les nouveaux descripteurs. De même, pour les solutions (a), (b) et (c) la majorité des internautes qui choisissaient de l'intermédiaire avant la modification, choisiraient de l'intermédiaire après la modification. D'autre part, ces solutions indiquent que les choisissant du Tiers (T) ou du Tiers++ (T++) avant la modification choisiraient de nouveau un niveau de garantie lié au Tiers (T, T+, T++) avec les nouveau descripteurs. Cependant, une minorité d'internautes ayant choisis du Tiers++ avant la modification, choisiraient plutôt

du Tous risque. La principale différence entre les stratégies EXBIC et AGBIC concerne la modalité ajoutée Tous Risque+ (TR+), où selon la stratégie EXBIC, les internautes choisissant cette modalité seraient ceux qui choisissaient la modalité Intermédiaire (I) auparavant alors que pour la stratégie AGBIC, ce sont des internautes qui choisissaient du Tiers (T) auparavant. Le Bayesian model averaging sur les modèles retournés par la stratégie AGBIC dispersent beaucoup plus les probabilités de transition que les autres et est beaucoup moins intuitif. D'un point de vue métier, les modèles (a), (b) et (c) sont tous trois interprétables et semblent cohérents.

x\y	T	T+	T++	I	TR	TR+	TR++
T	0.64	.	0.36	.	.	.	.
T++	.	0.90	.	.	.	.	0.10
I	.	.	.	0.74	.	0.26	.
TR	.	.	.	.	1.00	.	.

(a) EXBIC

x\y	T	T+	T++	I	TR	TR+	TR++
T	0.64	.	0.34	0.02	.	.	.
T++	.	0.90	.	.	.	.	0.10
I	.	.	0.03	0.71	.	0.26	.
TR	.	.	.	.	1.00	.	.

(b) Bayesian averaging EXBIC

x\y	T	T+	T++	I	TR	TR+	TR++
T	0.48	.	0.36	.	.	0.14	0.02
T++	0.62	0.28	.	.	0.1	.	.
I	.	0.23	.	0.76	.	.	.
TR	.	.	.	.	1.00	.	.

(c) AGBIC

x\y	T	T+	T++	I	TR	TR+	TR++
T	0.3	0.08	0.2	0.18	0.2	0.01	0.01
T++	0.27	0.06	0	0.13	0.3	0.2	0.01
I	0.17	0.11	0.15	0.23	0.2	0.1	0.01
TR	0.23	0.05	0.1	0.01	0.5	0.02	0.006

(d) Bayesian averaging AGBIC

Tableau 6.7 – Probabilités d'appariement estimées selon la stratégie utilisée, pour la variable NGS1.

NGS2 : La modification de cette variable repose uniquement sur la suppression de la modalité Tiers+ (T). Les modèles paramètres estimées pour les autres modalités devraient alors être très proche de 1. Le tableau 6.8 présente les modèles retournés par les stratégies EXBIC, AGBIC et les Bayesian Model Averaging (BMA) associés, pour ce jeu de données. Trois des quatre solutions (tableau (a), (b) et (c)), sont en accord pour estimer que 100% des internautes choisissant la modalité supprimée (T+) avant la modification choisiraient désormais la modalité (TR++). La stratégie EXBIC sélectionne le modèle le plus intuitif, où les probabilités estimées pour l'ensemble des modalités non modifiés sont estimées à 1. Le BMA associé à la stratégie suit ce modèle, mis à part pour la modalité (T++). La stratégie AGBIC retourne un modèle moins intuitif, notamment sur les modalités (T+) et (T++) mais répartie un plus les internautes. A l'instar de la variable précédente, le BMA associé à la stratégie AGBIC disperse beaucoup les internautes. Cependant,

la majorité des internautes choisissant les modalités (T, T+ et T++) avant la modification, sont répartis sur les modalités (T, T++ et I) après la modification. De même, la majorité des internautes choisissant les modalités (TR et TR++) avant la modification, sont répartis sur les modalités (TR, TR+ et TR++) après la modification.

$X\backslash Y$	T	T++	Inter	TR	TR+	TR++
T	1	.	.	.	.	.
T+	.	.	.	.	.	1
T++	.	1	.	.	.	.
Inter	.	.	1	.	.	.
TR	.	.	.	1	.	.
TR+	.	.	.	.	1	.
TR++	.	.	.	.	.	1

(a) EXBIC

$X\backslash Y$	T	T++	Inter	TR	TR+	TR++
T	1	.	.	.	.	.
T+	.	.	.	.	.	1
T++	.	.	.	.	.	1
Inter	.	.	0.99	.	0.01	.
TR	.	.	.	1	.	.
TR+	.	.	0.01	.	0.99	.
TR++	.	0.08	.	.	.	0.92

(b) BMA_EXBIC

$X\backslash Y$	T	T++	Inter	TR	TR+	TR++
T	0.87	.	0.13	.	.	.
T+	.	.	.	.	.	1
T++	.	.	.	1	.	.
Inter	.	.	1	.	.	.
TR	.	0.5	.	0.5	.	.
TR+	.	.	.	0.03	0.97	.
TR++	0.23	.	.	0.4	.	0.4

(c) AGBIC

$X\backslash Y$	T	T++	Inter	TR	TR+	TR++
T	0.35	0.11	0.16	0.008	0.26	0.11
T+	0.25	0.23	0.34	0.11	.	0.2
T++	0.19	0.34	0.21	0.09	0.16	.
Inter	0.11	0.18	0.12	0.34	0.23	.
TR	.	0.096	0.19	0.67	0	0.04
TR+	0.3	0.19	0.19	0.14	0.28	0.09
TR++	0.37	0.1	.	0.26	0.18	0.17

(d) BMA_AGBIC

Tableau 6.8 – Probabilités d'appariement estimées, pour la variable NGS2, selon la stratégie utilisée.

NGS3 : Pour cette variable, les modalités (T++ et TR+) ont été supprimées. De même que pour la modification précédente, il semblerait logique que les probabilités de transition pour les modalités non modifiées soient estimées à 1. Le tableau 6.9 indique les modèles retournés par les stratégies EXBIC, AGBIC et les Bayesian Model Averaging (BMA) associés, pour ce jeu de données. Les stratégies EXBIC et le BMA associé indiquent des modèles identiques et très intuitifs. Selon ces modèles, les internautes choisissant auparavant la modalité TR++ choisissent désormais la modalité TR+, ce qui parait logique. Les quatre solutions (tableau (a), (b), (c) et (d)), indiquent que les internautes ayant choisissant la modalité T++ avant la modification choisiraient les modalités (TR) ou (TR+). Pour cette variable, les quatre solutions sont intuitives. Les paramètres en rouge pour le tableau (d) montrent que le BMA associé à la stratégie AGBIC, bien que dispersant plus les

internautes que les autres modèles reste intuitif.

$X \backslash Y$	T	Inter	TR	TR+
T	1	.	.	.
T++	.	.	.	1
Inter	.	1	.	.
TR	.	.	1	.
TR+	.	.	.	1
TR++	.	.	.	1

(a) EXBIC

$X \backslash Y$	T	Inter	TR	TR+
T	1	.	.	.
T++	.	.	.	1
Inter	.	1	.	.
TR	.	.	1	.
TR+	.	.	.	1
TR++	.	.	.	1

(b) Bayesian averaging EXBIC

$X \backslash Y$	T	Inter	TR	TR+
T	1	.	.	.
T++	0.34	.	0.66	.
Inter	.	1	.	.
TR	.	0.05	0.95	.
TR+	.	0.3	.	0.7
TR++	.	.	1	.

(c) AGBIC

$X \backslash Y$	T	Inter	TR	TR+
T	0.7	0.05	0.25	0
T++	0.16	0.14	0.48	0.21
Inter	0.21	0.57	0.31	0.03
TR	0.18	0.21	0.55	0.05
TR+	0.13	0.1	0.5	0.27
TR++	0.28	0.31	0.3	0.1

(d) Bayesian averaging AGBIC

TABLEAU 6.9 – Probabilités d'appariement estimées, pour la variable NGS3, selon la stratégie utilisée.

CS : Le tableau 6.10 indique les modèles retournés par les stratégies EXBIC, AGBIC et les Bayesian Model Averaging (BMA) associés pour la variables "Statut du conducteur". Les solutions des tableaux (a), (b) et (c) indiquent de modèles similaires. Ces trois modèles indiquent que 100% des internautes ayant choisis la modalité "non" avant la modification, choisiraient de nouveau la modalité "non", s'ils revenaient avec les nouveaux descripteurs. De même, pour ces trois solutions 93% des internautes qui choisissaient "CP" avant la modification, choisiraient la modalité "CP" après la modification. De même que pour les variables précédentes, le Bayesian Model Averaging sur la stratégie AGBIC répartie beaucoup plus les probabilités de transition et semble un peu moins intuitif. Néanmoins, les quatre solutions indiquent que 90% des internautes choisissant la modalité "CP" avant la modification, la choisirait de nouveau et la majorité des internautes choisissant la modalité "Non" avant la modification, la choisirait de nouveau après la modification.

$X\backslash Y$	ACP	ACS	Non	CP	CS
CP	0.05	0.02	.	0.93	.
CS	.	.	.	.	1
Non	.	.	1	.	.

(a) EXBIC

$X\backslash Y$	ACP	ACS	Non	CP	CS
CP	0.04	0.03	.	0.93	.
CS	0.15	0.05	.	.	0.8
Non	.	.	1	.	.

(b) Bayesian Averging EXBIC

$X\backslash Y$	ACP	ACS	Non	CP	CS
CP	0.06	.	0.01	0.93	.
CS	.	0.24	.	.	0.76
Non	.	.	1	.	.

(c) AGBIC

$X\backslash Y$	ACP	ACS	Non	CP	CS
CP	0.005	0.005	0.07	0.9	0.02
CS	0.4	0.12	0.13	0.02	0.33
Non	0.061	0.02	0.7	0.15	0.05

(d) Bayesian Averging AGBIC

Tableau 6.10 – Probabilités d'appariement estimées, pour la variable CS, selon la stratégie utilisée.

6.7.2 Classification

L'objectif principal de la société MeilleureAssurance est de pouvoir prédire la ou les offres correspondant au profil et aux attentes des internautes. Dans cette partie, un modèle prédictif est créé. Le but de ce modèle est de prédire les internautes susceptibles de faire une MER pour une assurance automobile, donc intéressés par une des offres proposées. Pour ce modèle, 25 variables sont utilisées dont 20 sont catégorielles et 5 sont continues. La variable à prédire est une variable à deux modalités : Fiche/MER. Pour prédire cette variable, une régression logistique est utilisée. La régression logistique est réalisée avec l'outil "Dataiku [2]". L'objectif de cette partie est de comparer les performances du modèle suite aux différentes modifications de descripteurs des variables. Au fil du temps, quatre modifications ont été réalisées. A chaque modification, le modèle doit être récréé. La section suivante compare les performances du modèle créé 2 jours après une modification et les performances de modèle utilisant une étape de transition avec les stratégies AGBIC, EXBIC BMAEX et BMAAG. Les modèles sont comparés par rapport au taux d'erreur et à leurs capacités à prédire les vrais positifs (VP), pour chacune des modifications.

Données : Les données proviennent de la société MeilleureAssurance.com. Pour chaque modification, deux jeux de données ont été créés. Le premier jeu de données correspond aux données récoltées 1 mois avant la modification (AvM). Le second jeu de données correspond aux données récoltées sur 2 jours après la modification (ApM). Un troisième jeu de données (App) correspond à 1 mois de

2. https ://www.dataiku.com/

données récoltées après la modification de la variable sur lequel les modèles sont testés.

Protocole : Pour chaque modification, 5 modèles sont créés. Le premier modèle (ST) est un modèle sans étape de transition. Il est créé uniquement sur le jeu de données ApM. Pour les quatre autres modèles, le jeu de données AvM est modifié selon le modèle utilisé et les probabilités de transition estimées. Les jeux de données AvM et ApM sont ensuite réunis pour créer le modèle prédictif. Pour chacun des modèles, une régression logistique est utilisée. Pour chaque modèle le jeu de données utilisé est séparé en deux jeux de données. 70% du jeu de données est consacré à l'apprentissage du modèle et 30% des données sont consacrées au test du modèle. Chaque modèle de classification, pour chaque modification, est ensuite appliqué sur le jeu de donnée App. Les performances des modèles de classification sont alors comparées.

6.7.3 Comparaison

Pour évaluer la pertinence de notre modèle face aux modifications dans un modèle prédictif nous comparons les résultats de la classification d'un modèle sans transformation, où l'échantillon d'apprentissage est réinitialisé à chaque modification, aux résultats d'un modèle où la transition a été calculée et effectuée. Le tableau 6.11 présente les résultats de la classification pour la variable "Niveau de garantie souhaité". Le constat principal est que, dès le deuxième jour après la modification, le modèle avec étape transitoire est stable, comme il est possible de le constater avec le tableau 6.11(a). A contrario, il faut au minimum 10 jours au modèle sans étape transitoire pour atteindre le même taux d'erreur, tel qu'on peut le constater avec le tableau 6.11(b). Le même constat peut être fait pour le taux de prédiction des vrais positifs, tel qu'indiqué en gras dans le tableau 6.11(b). Pour chacune des modifications, le modèle sans étape de transition est surpassé par les modèles ayant une étape de transition. Cette étape est donc nécessaire pour avoir un modèle performant en peu de temps. Concernant les différentes stratégies utilisées, celles-ci indiquent des résultats similaires en termes d'amélioration des performances dans la classification finale.

6.8 Conclusion

L'objectif de ce chapitre était de trouver la stratégie permettant le meilleur compromis "résultats-rapidité". Pour cela 16 jeux de données simulés ont été générés et 4 jeux

	1j	2j	3j	6j	10j	1mois
Tx d'err	.	40%	42%	40%	35%	37%
VP	.	48%	44%	58%	56%	60%
VN	.	65%	64%	60%	66%	67%
App	.	112	175	414	593	1565

(a) Résultats de la classification sans l'étape transitoire.

	1j	2j	3j	6j	10j	1mois
Tx d'err	.	37%	37%	37%	37%	37%
VP	.	**61%**	**61%**	**61%**	**61%**	**61%**
VN	.	65%	64%	64%	64%	64%
App	.	12249	12312	12551	12730	13702

(b) Résultats de la classification avec l'étape transitoire et la méthode EXBIC.

Tableau 6.11 – Résultats la variable NGS avec la modification du 09/09/2014.

de données réelles, provenant de la société MeilleureAssurance.com ont été utilisés. Une première comparaison des critères de sélection de modèles a alors été réalisée. L'objectif de cette comparaison est d'étudier dans qu'elle mesure le critère BIC peut remplacer l'utilisation du critère BIL. Le critère BIL a été conçu pour sélectionner le "vrai" modèle, quel que soit le jeu de données. Cependant, son temps de calcul est très coûteux, alors que celui du critère BIC est très rapide. Pour la plupart des modèles, le critère BIC est aussi performant que le critère BIL. Néanmoins, sur de petits jeux de données, de taille $n < 500$, il est nécessaire d'utiliser le critère BIL. Suite à ces résultats, le critère BIC a été utilisé pour effectuer les autres comparaisons. Les méthodes **EXsearch** et **AGsearch** ont alors été comparées en fonction de la valeur minimisant le critère BIC retourné, leurs temps d'exécution et la précision de la valeur des paramètres estimés. La méthode **EXsearch** étant une méthode exhaustive, la valeur minimisant le critère BIC correspond à la valeur du modèle simulé et sert de référence. Les valeurs minimisant le critère BIC de la stratégie AGBIC ont alors été comparées aux valeurs de la stratégie EXBIC. La stratégie AGBIC, ayant des valeurs de critère BIC similaires à celles de la stratégie EXBIC, est aussi performante que la stratégie EXBIC sur les jeux de données simulés. D'autre part, la stratégie AGBIC, comparant plus de modèles, a sélectionné, sur les jeux de données réels, des modèles ayant une valeur critère BIC inférieure à celle du critère BIC de la stratégie EXBIC, donc un modèle "meilleur" au sens du critère BIC. En terme de temps de calcul, la stratégie AGBIC est beaucoup plus rapide que la stratégie EXBIC sur l'ensemble des jeux de données utilisés. La méthode **EXsearch** dépendant du nombre de modèles à comparer et de la taille du jeu de données, peut vite devenir

très coûteuse en temps de calcul. A titre d'exemple, il lui faut 7h pour comparer 4 095 modèles sur un échantillon de 15000 données alors que la stratégie AGBIC met 2min10. Cela est dû au nombre d'itération nécessaire à l'algorithme EM. La méthode **AGsearch** ne dépend que du nombre maximum d'itérations et de la taille de la population déterminée au préalable. Elle peut alors être beaucoup plus rapide. Concernant l'estimation des paramètres, les deux méthodes estiment des valeurs de paramètres très proches des valeurs simulées. Cependant, la stratégie AGBIC, ayant plus grande liberté sur le choix des paramètres estimés, peut-être plus précise pour certains modèles. Dans un dernier temps, les stratégies ont été comparées selon les objectifs initiaux de la société MeilleureAssurance.com. C'est-à-dire, l'interprétabilité des modèles et leur capacité à améliorer un modèle de classification. Pour élargir la famille de modèle, un Bayesian Model Averaging a été également effectué sur les ensembles de modèles des stratégies EXBIC et AGBIC. En terme de classification, l'ensemble des modèles améliore le modèle de classification sans étape de transition et ont des performances similaires. En terme d'interprétation, les stratégies EXBIC, AGBIC et le Bayesian Model Averaging appliqué à la stratégie EXBIC donnent des modèles intuitifs d'un point de vue métier. A l'inverse, le Bayesian Model Averaging appliqué à la stratégie AGBIC dispersent beaucoup plus les données et le modèle retourné est alors beaucoup moins intuitif d'un point de vue métier.

Conclusion générale et perspectives

Conclusion

Le domaine de la comparaison d'assurances implique de travailler avec des données évoluant constamment. En effet, pour répondre aux attentes de l'entreprise, les formulaires en ligne, d'où proviennent les données, sont régulièrement modifiés. Ces modifications constantes des variables et des descripteurs de variables complexifient les différentes analyses de la société. Dans ce travail, une modélisation probabiliste, basée sur la loi jointe des données observées a été proposée. Afin d'obtenir des modèles identifiables, des contraintes très simples ont été proposées. Ces contraintes imposent que certaines probabilités de transition soient fixées à zéro. Cette modélisation est présentée dans le chapitre 3. Les contraintes imposées sur les modèles impliquent de travailler avec un ensemble de modèles dont il est nécessaire d'estimer les paramètres. Le problème comportant des données manquantes, il a été décidé d'utiliser un algorithme EM pour réaliser l'estimation. Suite à l'estimation des paramètres, une méthode de sélection de modèles est nécessaire. Deux critères ont alors été proposés : un critère asymptotique, le critère BIC et un critère non asymptotique, le critère BIL. Dans le chapitre 4, il est montré que les limites du critère BIC peuvent être atteintes rapidement dans le cadre de notre problème, dû notamment à notre modélisation. Le critère BIL est un critère non asymptotique permettant de travailler sur des échantillons de petites tailles. Il repose sur le calcul de la vraisemblance intégrée, approchée en deux étapes. La première étape repose sur l'intégration exact de la distribution des données complètes sur les paramètres. La seconde étape repose sur une approximation de la somme de l'ensemble des valeurs possibles pouvant être prises par les individus $\mathbf{x}^+$. L'approximation de la somme est réalisée à travers une stratégie d'échantillonnage préférentiel et un échantillonneur de Gibbs. Bien qu'il n'y ait aucune garantie théorique concernant le fonctionnement du critère BIC lorsque les paramètres sont estimés sur le bords de leur espace, le critère BIC reste efficace sur les différentes simulations réalisées. A l'inverse,

sur de petits échantillons $n < 500$, le critère BIC a des difficultés à sélectionner le "quasi-vrai" modèle, contrairement au critère BIL qui atteint le comportement asymptotique très rapidement ($n = 20$). Le temps de calcul du critère BIL étant très coûteux en temps, dû aux différentes stratégies utilisées, le choix du critère BIC est privilégié pour des tailles d'échantillons suffisantes. Ce premier travail, utilisant différents outils statistiques, nécessite l'estimation et la comparaison de l'ensemble des modèles disponibles et correspond donc à une méthode exhaustive, notée **EXsearch**. Selon la taille de l'espace des modèles, inhérents au nombre de modalités des variables, le problème combinatoire peut rapidement devenir conséquent et la méthode **EXsearch** devient rapidement très consommatrice en temps, notamment à cause de l'algorithme EM. Une seconde méthode (**AGsearch**), reposant sur une méthode d'optimisation stochastique est alors proposée. L'objectif de la méthode **AGsearch** est de surpasser les défauts de la méthode **EXsearch**. Pour la méthode **AGsearch**, il a été choisi d'utiliser un algorithme génétique d'états stationnaires, permettant de trouver une "bonne" solution rapidement. L'algorithme génétique utilisé repose sur la modélisation réalisée précédemment et a été adapté pour répondre aux contraintes imposées sur les modèles. Dans notre algorithme génétique, une solution correspond à un modèle, et est composée de paramètres estimés étant des probabilités de transition et de paramètres fixés à zéro. La gestion des paramètres fixés à zéro est réalisée à l'aide d'un nouvel opérateur de croisement, proposé pour garder certaines informations. Ce nouvel opérateur repose sur l'adaptation d'opérateurs classiques (croisement uniforme et croisement binaire simulé) afin de n'être appliqué que sur les paramètres estimés. Cet opérateur est ensuite comparé à son homologue où l'application est possible sur tous les paramètres. Afin d'obtenir la méta-heuristique la plus efficace, une comparaison des 8 méta-heuristiques a été réalisée. Étant donné que chacune des solutions correspond à un modèle probabiliste, un opérateur de correction a également été créé et est appliqué après chaque opérateur de croisement et de mutation. L'opérateur de correction a pour but de pondérer les valeurs estimées par l'algorithme génétique afin que la somme des probabilités pour chacune des modalités de $\boldsymbol{x}$ soit égale à 1. Les 8 méta-heuristiques sont comparées avec un test de Kruskall Wallis puis par un test de Mann Whitney. Selon les tests de comparaisons effectués, il résulte que l'algorithme génétique est plus efficace lorsque le nouvel opérateur de croisement est utilisé. Le principal objectif de la méthode **AGsearch** est de challenger les résultats de la méthode **EXsearch**. Pour cela, les deux méthodes ont été comparées à travers différents critères. Deux critères de sélection de modèles ont été proposés et peuvent être utilisés, combinés aux méthodes **AGsearch** et **EXsearch**, quatre stratégies sont finalement comparées. Dans un premier temps, les performances des deux critères de sélection de

modèles sont comparés sur 16 jeux de données simulées. Cela nous permet de constater que le critère BIC a des performances similaires au critère BIL sur des jeux de données de taille $n > 500$. Cependant, sur des jeux de données de taille $n < 500$, il est nécessaire d'utiliser le critère BIL. La plupart des jeux de données étant de taille suffisante, le critère BIC est utilisé dans la suite des comparaisons. Les méthodes **AGsearch** et **EXsearch** sont ensuite comparées, selon leurs valeurs minimisant le critère BIC, pour les 16 jeux de données simulées et les 4 jeux de données réelles. Les résultats des valeurs minimisant le critère BIC sont similaires, pour les deux méthodes, sur les jeux de données simulés. La méthode **AGsearch** est donc aussi performante que la méthode **EXsearch**, qui nous sert de référence. Cela implique qu'une méthode exhaustive n'est pas forcément nécessaire. De plus, la méthode **AGsearch** permet d'élargir l'espace de modèle comparé. De ce fait, sur les jeux de données réelles, la méthode **AGsearch** permet de trouver potentiellement un modèle meilleur, au sens du critère BIC, que celui retourné par la méthode **EXsearch**. Suite à cette comparaison, les temps d'exécutions des deux méthodes ont également été comparés, l'objectif de la méthode **AGsearch** étant principalement de gagner en rapidité. Le nombre d'itérations nécessaires à la convergence de l'algorithme EM implique un temps de calcul coûteux, notamment lorsque l'espace de modèles est grand. L'algorithme génétique ne dépendant pas de la taille de l'espace de modèles, il est plus rapide sur l'ensemble des jeux de données utilisés. Plus l'espace de modèles est grand, plus l'écart entre les temps d'exécution des deux méthodes est important. Par exemple, sur les jeux de données de taille $n = 5000$ et $\mathbf{\Delta} = 16807$, la méthode **AGsearch** est 800 fois plus rapide que la méthode **EXsearch**, alors que sur les jeux de données de taille $n = 10000$ et $\mathbf{\Delta} = 26$, la méthode **AGsearch** est 4.5 fois plus rapide que la méthode **EXsearch**. Sur les deux critères principaux, la méthode **AGsearch** se révèle plus efficace que la méthode **EXsearch**. La qualité de l'estimation a également été étudiée et les deux méthodes donnent des résultats équivalents. Initialement, la société MeilleureAssurance.com avait deux objectifs principaux étant : Avoir un modèle prédictif robuste prenant en compte les modifications de descripteurs de variable et pouvoir interpréter ces modèles. Afin d'évaluer la pertinence de ce travail pour la société, les deux méthodes ont également été comparées sur ces deux axes. De plus, les deux méthodes retournant des modèles très proche en terme de critère BIC, une stratégie de Bayesian Model Averaging est également effectuée sur chacune des deux méthodes et les modèles agrégés sont étudiés en terme d'interprétabilité et d'amélioration des performances d'un modèle de classification. Sur les quatre jeux de données réelles testés, le modèle le moins intuitif est le modèle agrégé par le Bayesian Model Averaging de la méthode **AGsearch**. Le modèle le plus intuitif est le modèle sélectionné par la méthode **EXsearch**. Le modèle sélectionné par méthode

AGsearch donne des résultats plus surprenant, bien que restant cohérent. Concernant la classification, l'étape de transition effectuée par l'une des méthodes proposées est nécessaire. Cependant, les quatre modèles comparés améliorent les performances de la classification de manière similaire.

Dans ce travail, deux méthodes ont été proposées pour l'estimation et la sélection de modèles. La méthode ayant le meilleur compromis "résultats-rapidité" est la méthode **AGsearch** associé au critère de sélection BIC. Cependant, il a été montré que le critère n'était pas performant pour des échantillons de taille $n < 500$, pour ces échantillons il est nécessaire d'utiliser le critère BIL.

Application chez MeilleureAssurance.com

La méthode proposée trouve diverses applications au sein de la société *MeilleureAssurance.com*. D'un point de vue expérience utilisateur, le dernier changement de la variable "Niveau de garantie souhaité" avait pour but de regrouper certains descripteurs de la variable pour améliorer l'affichage des offres en page de restitution. L'intuition sous-jacente étant que les internautes choisissant du Tiers++ se tourneraient vers du Tiers. Or, avec les modèles de la figure 6.9 on constate que cette intuition n'est pas vérifiée. La méthode proposée peut également concerner les différents outils mis à disposition des internautes. Notamment, les outils tels que le baromètre [3], calculant des tarifs moyens selon différentes variables. Lorsque les descripteurs d'une variable changent, tel que pour la variable "niveau de garantie souhaité", il est alors nécessaire d'appliquer la méthode pour garder des résultats fiables et cohérents d'un mois à l'autre. Il en est de même pour la majorité des études réalisées au sein de la société *MeilleureAssurance.com*. Enfin, la dernière application au sein de la société est l'utilisation de la méthode dans un modèle prédictif, notamment pour prédire les internautes les plus intéressés ou susceptibles de faire une MER.

Perspectives

Perspectives probabilistes

Clustering : Dans ce travail, la modélisation est réalisée sur l'ensemble de la population de l'échantillon. Dans la pratique, il est possible que l'échantillon contienne

3. https ://assurance.meilleurtaux.com/barometre

différentes typologies d'individus. L'objectif serait alors d'adapter le modèle en fonction du groupe d'individus sur lequel il est effectué.

Modèle utilisant d'autres contraintes : Les contraintes de type binaire utilisées pour le modèle sont très fortes. Une perspective serait de proposer d'autres contraintes, respectant l'identifiabilité en paramètres. Les nouvelles contraintes pourraient être l'utilisation des modes pour chaque modalité de $\boldsymbol{x}$ ou l'utilisation de coefficients exponentiels.

Relâchement des contraintes des modèles : Afin d'avoir des modèles respectant les données de départ, ceux-ci ont été contraints pour conserver toujours le même nombre de modalités p et q. Une perspective serait de relâcher ces contraintes pour élargir la famille de modèles. Il sera nécessaire de s'assurer que les modèles avec ces contraintes relâchées restent identifiables en paramètres.

Prise en compte rapide de l'ajout d'une variable : Il est possible que le comparateur d'assurances soit amené à ajouter des questions aux formulaires. Une perspective serait alors d'avoir un modèle de classification qui s'adapte très rapidement à ces ajouts. C'est-à-dire qu'avec très peu de données, le modèle initial soit capable d'inclure, si nécessaire, les données de la variable ajoutée. L'objectif étant d'avoir un modèle s'adaptant aux changements rapidement plutôt que de devoir recréer un nouveau modèle comprenant les nouvelles données provenant des ajouts de questions.

Perspectives recherche opérationnelle

Actuellement, les opérateurs choisis pour l'algorithme génétique sont des opérateurs usuels et naïfs. Une perspective de ce travail serait d'ajouter plus d'intelligence dans ces opérateurs. Par exemple, la création de nouveaux opérateurs utilisant des connaissances statistiques tel que les lois des distributions pourrait permettre d'améliorer les résultats de notre algorithme.

Bibliographie

[1] R. B. Agrawal, K. Deb et R. Agrawal, « Simulated binary crossover for continuous search space », *Complex systems*, t. 9, n° 2, p. 115–148, 1995.

[2] H. Akaike, « A new look at the statistical model identification », *IEEE transactions on automatic control*, t. 19, n° 6, p. 716–723, 1974.

[3] A. Almaksour et E. Anquetil, « Apprentissage incrémental en-ligne pour des problèmes de classification évolutifs », 2010.

[4] J. A. Anderson et S. C. Richardson, « Logistic discrimination and bias correction in maximum likelihood estimation », *Technometrics*, t. 21, n° 1, p. 71–78, 1979.

[5] R. K. Ando et T. Zhang, « A framework for learning predictive structures from multiple tasks and unlabeled data », *Journal of Machine Learning Research*, t. 6, n° Nov, p. 1817–1853, 2005.

[6] A. Arnold, R. Nallapati et W. Cohen, « A comparative study of methods for transductive transfer learning », in *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, IEEE, 2007, p. 77–82.

[7] K. G. Balcombe, « Model selection using information criteria and genetic algorithms », *Computational Economics*, t. 25, n° 3, p. 207–228, 2005.

[8] A. L. Bedenel, C. Biernacki et L. Jourdan, « Appariement de descripteurs évoluant en temps », in *48èmes Journées des Statistiques Française*, 2016.

[9] A. L. Bedenel, L. Jourdan et C. Biernacki, « Probabilities estimation by a genetic algorithm », fév. 2018. adresse : https://hal.archives-ouvertes.fr/hal-01868195.

[10] A. L. Bedenel, L. Jourdan et C. Biernacki, « Probability estimation by an adapted genetic algorithm in web insurance », in *Learning and Intelligent Optimization Conference (LION 12)*, Kalamata, Greece, juin 2018. adresse : https://hal.archives-ouvertes.fr/hal-01885117.

[11] F. Beninel, C. Biernacki, C. Bouveyron, J. Jacques et A. Lourme, « Parametric link models for knowledge tranfer in statistical learning. », *Knowledge transfert ; Practices, Types and Challenges*, 2012.

[12] C. Biernacki, G. Celeux, A. Echenim, G. Govaert et F. Langrognet, « Le logiciel MIXMOD d'analyse de mélange pour la classification et l'analyse discriminante », *La revue de Modulad*, n° 35, p. 25–44, 2006.

[13] C. Biernacki, G. Celeux et G. Govaert, « Exact and Monte Carlo calculations of integrated likelihoods for the latent class model », *Journal of Statistical Planning and Inference*, t. 140, n° 11, p. 2991–3002, 2010.

[14] R. R. Bies, M. Muldoon, B. G. Pollock, S. Manuck, G. Smith et M. E. Sale, « A genetic algorithm-based, hybrid machine learning approach to model selection », *Journal of Pharmacokinetics and Pharmacodynamics*, t. 33, n° 2, p. 195–221, 2006.

[15] A. Blauth et I. Pigeot, « Using Genetic Algorithms for Model Selection in Graphical Models », 2002.

[16] J. Blitzer, R. McDonald et F. Pereira, « Domain adaptation with structural correspondence learning », in *Proceedings of the 2006 conference on empirical methods in natural language processing*, Association for Computational Linguistics, 2006, p. 120–128.

[17] M. Bouillon, E. Anquetil et A. Almaksour, « Apprentissage incrémental et décrémental », 2012.

[18] C. Bouveyron, P. Gaubert et J. Jacques, « Adaptive models in regression for modeling and understanding evolving populations », *Journal of Case Studies in Business, Industry and Government Statistics (CSBIGS)*, t. 4, n° 2, p. 83–92, 2011.

[19] C. Bouveyron et J. Jacques, « Adaptive mixtures of regressions : Improving predictive inference when population has changed », *Communications in Statistics-Simulation and Computation*, t. 43, n° 10, p. 2570–2592, 2014.

[20] L. Breiman, J. Friedman, R. Olshen et C. Stone, « Classification and Regression Trees (Wadsworth & Brooks Cole, Monterey, CA). A powerful yet simple technique for ecological data analysis », *Ecology*, t. 81, p. 3178–3192, 1984.

[21] G. Casella, C. P. Robert et M. Wells, « Mixture models, latent variables and partitioned importance sampling », *Statistical Methodology*, t. 1, n° 1-2, p. 1–18, 2004.

[22] B. Charles, « Modélisation et classification des données de grande dimension : application à l'analyse d'images. », thèse de doct., Université Joseph-Fourier-Grenoble I, 2006.

[23] D. R. Cox et N. Reid, « Parameter orthogonality and approximate conditional inference », *Journal of the Royal Statistical Society. Series B (Methodological)*, p. 1–39, 1987.

[24] W. Dai, Y. Chen, G. Xue, Q. Yang et Y. Yu, « Translated learning : Transfer learning across different feature spaces », in *Advances in neural information processing systems*, 2009, p. 353–360.

[25] W. Dai, Q. Yang, G. Xue et Y. Yu, « Boosting for transfer learning », in *Proceedings of the 24th international conference on Machine learning*, ACM, 2007, p. 193–200.

[26] C. Darwin, *On the origin of species, 1859*. Routledge, 2004.

[27] C. Darwin, « On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life », *J. Murray.[MG] David, V. & Cazala, P.(2000) Anatomical and pharmacological specificity of the rewarding effect elicited by microinjections of morphine into the nucleus accumbens of mice. Psychopharmacology*, t. 150, p. 2434, 1860.

[28] V. B. Dasarathy, « Nearest neighbor ({NN}) norms :{NN} pattern classification techniques », 1991.

[29] K. Deb et D. Deb, « Analysing mutation schemes for real-parameter genetic algorithms », *International Journal of Artificial Intelligence and Soft Computing*, t. 4, n° 1, p. 1–28, 2014.

[30] A. P. Dempster, N. M. Laird et D. B. Rubin, « Maximum likelihood from incomplete data via the EM algorithm », *Journal of the Royal Statistical Society. Series B*, p. 1–38, 1977.

[31] L. Deng, D. Yu et al., « Deep learning : methods and applications », *Foundations and Trends® in Signal Processing*, t. 7, n° 3–4, p. 197–387, 2014.

[32] M. Dorigo, « Optimization, learning and natural algorithms », *PhD Thesis, Politecnico di Milano*, 1992.

[33] M. Dorigo, V. Maniezzo et A.Colorni, « Ant system : optimization by a colony of cooperating agents », *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, t. 26, n° 1, p. 29–41, 1996.

[34] J. Durillo et A. J. Nebro, « jMetal : A Java framework for multi-objective optimization », *Advances in Engineering Software*, t. 42, n° 10, p. 760–771, 2011.

[35] A. Eiben, J. Smith et al., *Introduction to evolutionary computing*. Springer, 2003, t. 53.

[36] K. D. Feuz et D. J. Cook, « Transfer learning across feature-rich heterogeneous feature spaces via feature-space remapping (FSR) », *ACM Transactions on Intelligent Systems and Technology (TIST)*, t. 6, n° 1, p. 3, 2015.

[37] R. Fisher, « The use of multiple measurements in taxonomic problems », *Annals of human genetics*, t. 7, n° 2, p. 179–188, 1936.

[38] L. J. Fogel, A. J. Owens et M. J. Walsh, « Artificial intelligence through simulated evolution », 1966.

[39] M. Fréchet, « Sur l'extension de certaines évaluations statistiques au cas de petits échantillons », *Revue de l'Institut International de Statistique*, p. 182–205, 1943.

[40] J. Friedman, T. Hastie et R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, t. 1.

[41] G. Fung et O. Mangasarian, « Incremental support vector machine classification », in *Proceedings of the 2002 SIAM International Conference on Data Mining*, SIAM, 2002, p. 247–260.

[42] J. Gao, F. Liang, W. Fan, Y. Sun et J. Han, « Graph-based consensus maximization among multiple supervised and unsupervised models », in *Advances in Neural Information Processing Systems*, 2009, p. 585–593.

[43] A. Gelfand et A. Smith, « Sampling-based approaches to calculating marginal densities », *Journal of the American statistical association*, t. 85, n° 410, p. 398–409, 1990.

[44] S. Geman et D. Geman, « Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images », *IEEE Transactions on pattern analysis and machine intelligence*, n° 6, p. 721–741, 1984.

[45] F. Glover, « Future paths for integer programming and links to artificial intelligence », *Computers & operations research*, t. 13, n° 5, p. 533–549, 1986.

[46] F. Glover, « Heuristics for integer programming using surrogate constraints », *Decision sciences*, t. 8, n° 1, p. 156–166, 1977.

[47] D. E. Goldberg, « Genetic algorithms in search, optimization, and machine learning/David E », *Goldberg.–[USA] : Addison-Wesley*, 1989.

[48] G. Govaert, *Data Analysis*. Wiley, 2009, chap. 8.

[49] G. Govaert, *Data Analysis*. Wiley, 2009, chap. 6.

[50] L. Hansen, « Large sample properties of generalized method of moments estimators », *Econometrica : Journal of the Econometric Society*, p. 1029–1054, 1982.

[51] P. Hansen, « The steepest ascent mildest descent heuristic for combinatorial programming », in *Congress on numerical methods in combinatorial optimization, Capri, Italy*, 1986, p. 70–145.

[52] M. Hasnat, J. Velcin, S. Bonnevay et J. Jacques, « Evolutionary clustering for categorical data using parametric links among multinomial mixture models », *Econometrics and Statistics*, t. 3, p. 141–159, 2017.

[53] T. Haveliwala, « Topic-sensitive pagerank : A context-sensitive ranking algorithm for web search », *IEEE transactions on knowledge and data engineering*, t. 15, n° 4, p. 784–796, 2003.

[54] J. J. Heckman, *Sample selection bias as a specification error (with an application to the estimation of labor supply functions)*, 1977.

[55] J. A. Hoeting, D. Madigan, A. E. Raftery et C. T. Volinsky, « Bayesian Model Averaging : A Tutorial », *Statistical Science*, t. 14, n° 4, p. 382–401, 1999.

[56] J. H. Holland, *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[57] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf et A. J. Smola, « Correcting sample selection bias by unlabeled data », in *Advances in neural information processing systems*, 2007, p. 601–608.

[58] J. Jacques et C. Biernacki, « Analyse discriminante sur données binaires lorsque les populations d'apprentissage et de test sont différentes. », in *DMAS*, 2005, p. 129.

[59] H. Jeffreys, « An invariant form for the prior probability in estimation problems », *Proc. R. Soc. Lond. A*, t. 186, n° 1007, p. 453–461, 1946.

[60] L. Jourdan, « Métaheuristiques pour l'extraction de connaissances : Application à la génomique », thèse de doct., Université des Sciences et Technologie de Lille-Lille I, 2003.

[61] J. Kennedy et R. C. Eberhart, « Particle swarm optimization », in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, t. 4, Perth, Australia, IEEE Service Center, Piscataway, NJ, 1995, p. 1942–1948.

[62] C. Keribin, « Les méthodes bayésiennes variationnelles et leur application en neuroimagerie : une étude de l'existant », thèse de doct., INRIA, 2009.

[63] C. Keribin, « Méthodes bayésiennes variationnelles : concepts et applications en neuroimagerie », *Journal de la Société Française de Statistique*, t. 151, n° 2, p. 107–131, 2010.

[64] S. Kirkpatrick, C. D. Gelatt et M. P. Vecchi, « Optimization by simulated annealing », *science*, t. 220, n° 4598, p. 671–680, 1983.

[65] J. R. Koza, *Genetic Programming II, Automatic Discovery of Reusable Subprograms*. MIT Press, Cambridge, MA, 1992.

[66] B. Krishnapuram, S. Yu et R. B. Rao, *Cost-sensitive Machine Learning*. CRC Press, 2011.

[67] B. Kulis, K. Saenko et T. Darrell, « What you saw is not what you get : Domain adaptation using asymmetric kernel transforms », in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, 2011, p. 1785–1792.

[68] E. Lebarbier et T. Mary-Huard, « Une introduction au critère BIC : fondements théoriques et interprétation », *Journal de la Société française de statistique*, t. 147, n° 1, p. 39–57, 2006.

[69] F. Li, S. J. Pan, O. Jin, Q. Yang et X. Zhu, « Cross-domain co-extraction of sentiment and topic lexicons », in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics : Long Papers-Volume 1*, Association for Computational Linguistics, 2012, p. 410–419.

[70] M. Long, J. Wang, G. Ding, J. Sun et S. Y. Philip, « Transfer feature learning with joint distribution adaptation », in *Computer Vision (ICCV), 2013 IEEE International Conference on*, IEEE, 2013, p. 2200–2207.

[71] G. Loosli, S. Canu et L. Bottou, « SVM et apprentissage des très grandes bases de données », 2006.

[72] H. Lourenço, O. Martin et T. Stützle, « Iterated local search », in *Handbook of metaheuristics*, Springer, 2003, p. 320–353.

[73] A. Lourme et C. Biernacki, « Simultaneous Gaussian Model-Based Clustering for Sample of Multiple Origins », *Computational Statistics*, p. 371–391, 2013.

[74] J. MacQueen et al., « Some methods for classification and analysis of multivariate observations », in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Oakland, CA, USA, t. 1, 1967, p. 281–297.

[75] H. B. Mann et A. Wald, « On stochastic limit and order relationships », *The Annals of Mathematical Statistics*, t. 14, n° 3, p. 217–226, 1943.

[76] A. W. Marshall, « The use of multi-stage sampling schemes in Monte Carlo computations », RAND CORP SANTA MONICA CALIF, rapp. tech., 1954.

[77] O. Martin, S. Otto et E. W. Felten, « Large-step Markov chains for the traveling salesman problem », 1991.

[78] G. McLachlan et T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, t. 382.

[79] B. Miller, D. Goldberg et al., « Genetic algorithms, tournament selection, and the effects of noise », *Complex systems*, t. 9, n° 3, p. 193–212, 1995.

[80] S. A. Murphy et A. W. V. der Vaart, « On profile likelihood », *Journal of the American Statistical Association*, t. 95, n° 450, p. 449–465, 2000.

[81] M. K. Ng, Q. Wu et Y. Ye, « Co-transfer learning via joint transition probability graph based method », in *Proceedings of the 1st international workshop on cross domain knowledge discovery in web and social network mining*, ACM, 2012, p. 1–9.

[82] S. J. Pan et Q. Yang, « A survey on transfer learning », *IEEE Transactions on knowledge and data engineering*, t. 22, n° 10, p. 1345–1359, 2010.

[83] C. H. Papadimitriou, « The complexity of combinatorial optimization problems. », 1976.

[84] S. Paterlini et T. Minerva, « Regression model selection using genetic algorithms », in *Proceedings of the 11th WSEAS international conference on nural networks and 11th WSEAS international conference on evolutionary computing and 11th WSEAS international conference on Fuzzy systems*, 2010, p. 19–27.

[85] K. Pearson, « Contributions to the mathematical theory of evolution », *Philosophical Transactions of the Royal Society of London. A*, t. 185, p. 71–110, 1894.

[86] R. Polikar, L. Upda, S. S. Upda et V. Honavar, « Learn++ : An incremental learning algorithm for supervised neural networks », *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, t. 31, n° 4, p. 497–508, 2001.

[87] M. Poussevin, E. Guardia-Sebaoun, V. Guigue et P. Gallinari, « Recommandation par combinaison de filtrage collaboratif et d'analyse de sentiments », in *CORIA 2014-COnférence en Recherche det Applications*, 2014, p. 27–42.

[88] J. R. Quinlan, « C4. 5 : Programming for machine learning », *Morgan Kauffmann*, t. 38, p. 48, 1993.

[89] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer et N. D. Lawrence, *Dataset shift in machine learning*. The MIT Press, 2009.

[90] A. Raftery, « Bayesian model selection in social research », *Sociological methodology*, p. 111–163, 1995.

[91] I. Rechenberg, « Evolutionsstrategie : Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. frommann-holzbog, Stuttgart, 1973 », 1994.

[92] C. Reeves, « Genetic algorithms », in *Handbook of metaheuristics*, Springer, 2003, p. 55–82.

[93] C. Robert, *Le choix bayésien : Principes et pratique*. Springer Science & Business Media, 2006.

[94] C. Robert et G. Casella, *Méthodes de Monte-Carlo avec R*. Springer Science & Business Media, 2011.

[95] C. Robert et G. Casella, *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[96] S. Ruping, « Incremental learning with support vector machines », in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, IEEE, 2001, p. 641–642.

[97] K. Saenko, B. Kulis, M. Fritz et T. Darrell, « Adapting visual category models to new domains », in *European conference on computer vision*, Springer, 2010, p. 213–226.

[98] F. Santos, « Arbres de décision », 2015.

[99] J. C. Schlimmerand et D. Fisher, « A case study of incremental concept induction », in *AAAI*, t. 86, 1986, p. 496–501.

[100] G. Schwarz et al., « Estimating the dimension of a model », *The annals of statistics*, t. 6, n° 2, p. 461–464, 1978.

[101] H. Schwefel, *Numerical optimization of computer models*. John Wiley & Sons, Inc., 1981.

[102] I. Shafarevich et A. Remizov, *Linear algebra and geometry*. Springer Science & Business Media, 2012.

[103] C. E. Shannon, « A Mathematical Theory of Communication », *Bell System Technical Journal*, t. 27, n° 3, p. 379–423, doi : 10.1002/j.1538-7305.1948.tb01338.x. eprint : https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1948.tb01338.x. adresse : https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1948.tb01338.x.

[104] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau et M. Kawanabe, « Direct importance estimation with model selection and its application to covariate shift adaptation », in *Advances in neural information processing systems*, 2008, p. 1433–1440.

[105] N. A. Syed, H. Liu et K. Sung, « Handling concept drifts in incremental learning with support vector machines », in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 1999, p. 317–321.

[106] E. Talbi, *Metaheuristics : from design to implementation*. John Wiley & Sons, 2009, t. 74.

[107] T. Tommasi, F. Orabona et B. Caputo, « Learning categories from few examples with multi model knowledge transfer », *IEEE transactions on pattern analysis and machine intelligence*, t. 36, n° 5, p. 928–941, 2014.

[108] H. Tong, C. Faloutsos et J. Pan, « Random walk with restart : fast solutions and applications », *Knowledge and Information Systems*, t. 14, n° 3, p. 327–346, 2008.

[109] P. E. Utgoff, « Incremental induction of decision trees », *Machine learning*, t. 4, n° 2, p. 161–186, 1989.

[110] P. E. Utgoff, N. C. Berkman et J. A. Clouse, « Decision tree induction based on efficient tree restructuring », *Machine Learning*, t. 29, n° 1, p. 5–44, 1997.

[111] V. Vandewalle, « Les modèles de mélange, un outil utile pour la classification semi-supervisée », *Revue MODULAD*, p. 121–145, 2009.

[112] V. Vapnik, « The Nature of Statistical Learning Theory », 1995.

[113] F. Vavak et T. C. Fogarty, « A comparative study of steady state and generational genetic algorithms for use in nonstationary environments », in *AISB workshop on Evolutionary Computing*, Springer, 1996, p. 297–304.

[114] G. Venturini, « Algorithmes génétiques et apprentissage », *Revue d'intelligence artificielle*, t. 10, n° 2-3, p. 345–387, 1996.

[115] K. Weiss, T. M. Khoshgoftaar et D. Wang, « A survey of transfer learning », *Journal of Big Data*, t. 3, n° 1, p. 9, 2016.

[116] L. Yao et W. A. Sethares, « Nonlinear parameter estimation via the genetic algorithm », *IEEE Transactions on signal processing*, t. 42, n° 4, p. 927–935, 1994.

[117] J. T. Zhou, I. W. Tsang, S. J. Pan et M. Tan, « Heterogeneous domain adaptation for multiple classes », in *Artificial Intelligence and Statistics*, 2014, p. 1095–1103.

Annexe A

Identifiabilité en paramètres

Montrer qu'un modèle δ est identifiable en paramètre revient à résoudre l'équation en $p_{hh'}$:

$$\sum_{h=1}^{p} p_{hh'} p_h = q_{h'}, \quad \forall h' = 1, \ldots, q \tag{A.1}$$

$$s.c = \left\{ \begin{array}{r} \sum_{h=1}^{p} p_h = 1 \\ \sum_{h'=1}^{q} p_{hh'} = 1, \forall h \\ p_{hh'} = 0, \forall h' \in I_h, \text{ soit } \Delta_h \delta_{hh'} = \{0\} \end{array} \right.$$

où $\Delta_h\{\delta_{hh'}\}$ est l'ensemble des paramètres fixés à zéro. La problématique est alors de trouver l'ensemble des modèles identifiables possibles. Le système ainsi que les contraintes qui lui sont associées sont linéaires. De ce fait, la forme matricielle et ses propriétés peuvent être utilisées pour répondre à cette problématique.

Modélisation matricielle L'équation A.1 revient à un système linéaire où les paramètres inconnus sont les paramètres $\mathbf{p}_.$. Ce système peut alors être formalisé sous forme matricielle par : $\mathbf{A}_p \mathbf{p}_. = \mathbf{q}$, où $\mathbf{q} = (q_{h'})_{h'=1,\ldots,q}$ et $\mathbf{A}_p$ est une matrice de taille qm de la forme :

$$\mathbf{A}_p = \begin{pmatrix} p_1 & 0 & \ldots & 0_q & p_h & 0 & \ldots & 0_{hq} & p_p & 0 & \ldots & 0_m \\ 0 & p_1 & 0 & 0_q & 0 & p_h & 0 & 0_{hq} & 0 & p_p & 0 & 0_m \\ \vdots & 0 & \ddots & 0_q & \vdots & 0 & \ddots & 0_{hq} & \vdots & 0 & \ddots & 0_m \\ 0_{q_1} & 0 & 0 & p_1 & 0 & 0 & 0 & p_h & 0 & 0 & 0 & p_{p_m} \end{pmatrix}$$

où m correspond à la taille du vecteur $\mathbf{p}_.$: $m = qp$.

La contrainte $\sum_{h'=1}^{q} p_{hh'} = 1, \forall h$, liée à l'équation A.1 est ensuite modélisée par la matrice **C** de taille pm et est de la forme :

$$\mathbf{C} = \begin{pmatrix} \boxed{1_{11} \quad \cdots \quad 1_{1q}} & 0 & \\ 0 & \boxed{1_{h1} \quad \cdots \quad 1_{hq}} & 0 \\ & 0 & \boxed{1_{p1} \quad \cdots \quad 1_{pm}} \end{pmatrix}$$

Le modèle complet, sans contrainte peut alors être écrit sous la forme suivante :

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{C} \end{pmatrix} (\mathbf{p}_{.}) = \begin{pmatrix} \mathbf{q} \\ 1 \end{pmatrix} \tag{A.2}$$

Utilisant les propriétés des matrices, notamment les propriétés liées au rang, il est alors possible de définir si ce système, sans la contrainte des paramètres fixés à zéro, est identifiable.

Définition Le rang d'une matrice **A**, noté $R(\mathbf{A})$ correspond au nombre maximal de vecteurs lignes (ou colonnes) linéairement indépendants.

Théorème 1 Selon le théorème de **Rouché-Fonténé** [102], un système d'équations linéaires à m inconnues, de la forme $\mathbf{Ax} = \mathbf{b}$, possède une solution si et seulement si le rang de la matrice des coefficients **A** est égal à celui de la matrice augmentée $(\mathbf{A}|\mathbf{b})$.

Théorème 2 Selon le théorème de **Rouché-Fonténé** [102], lorsque le système d'équations linéaires admet une solution, celle-ci est **unique** si et seulement si le rang de la matrice **A** est égal au nombre de paramètres inconnus (soit m, la taille du vecteur $\mathbf{p}_{.}$).

Sans les contraintes imposées sur le modèle, le rang $R(\mathbf{A}_p)$ est inférieur à m, la solution admise par le système n'est donc pas unique et le modèle non contraint n'est pas identifiable. L'identifiabilité du modèle repose alors sur les contraintes binaires proposées dans la section 3.3.1 et sur le nombre de probabilités de transition fixées à zéro. Les contraintes de type binaire sont modélisées par la matrice Δ, présentées dans la section 3.3.2. Afin de garder des modèles répondant aux données initiales, c'est-à-dire avec un nombre de modalités p et q identiques pour chaque modèle, la matrice Δ doit respecter les contraintes suivantes :

— Il y a au moins un départ pour chaque modalité de $(\boldsymbol{x})_h \sum_{h'=1}^{q} (1 - \delta_{hh'}) \geq 1, \forall h$

— Il y a au moins une arrivée pour chaque modalité de $(\boldsymbol{y})_{h'} \sum_{h=1}^{p} (\delta_{hh'}) \geq 1, \forall h'$

Ces contraintes formalisent l'ensemble des modèles possibles selon les probabilités de transition fixées à zéro.

Objectif L'objectif est désormais de montrer qu'il existe une solution unique au système : $\mathbf{A}_p^{(\delta)} \times \mathbf{p}_{\cdot} = \mathbf{q}^{(\delta)}$, c'est-à-dire avoir une matrice $\mathbf{A}_p^{(\delta)}$ de plein rang, soit $R(\mathbf{A}_p^{(\delta)}) = m$, avec

$$\mathbf{A}_p^{(\delta)} = \begin{pmatrix} \mathbf{A}_p \\ \mathbf{C} \\ \Delta \end{pmatrix} \text{ et } \mathbf{q}^{(\delta)} = \begin{pmatrix} q_1 \\ \vdots \\ q_q \\ 1_1 \\ \vdots \\ 1_p \\ 0_1 \\ \vdots \\ 0_m \end{pmatrix}$$

Proposition Une condition nécessaire et suffisante pour que le système définisse une solution **unique** et donc avoir $R(\mathbf{A}_p^{(\delta)}) = m$ est :

$$\sum_{h'=1}^{q} \sum_{h=1}^{p} (1 - \delta_{hh'}) \geq m - (q + \dim(p)) \tag{A.3}$$

Condition nécessaire Une condition nécessaire pour avoir un modèle identifiable est que le nombre de probabilités de transition fixées à zéro soit compris dans les bornes suivantes :

$$\boxed{m - (q + \dim(p)) \leq \sum_{h=1}^{q} \sum_{i=1}^{p} (1 - \delta_{hh'}) \leq m - \max(p, q)} \tag{A.4}$$

Preuve Les lignes de la matrice $\mathbf{A}_p^{(\delta)}$ sont linéairement indépendantes, donc $R(\mathbf{A}_p^{(\delta)}) = q$. Si tous les passages sont possibles, c'est-à-dire que l'ensemble des $\delta_{hh'} = 1$, $\forall h, h'$, alors la matrice Δ correspond à une matrice nulle et la matrice $\mathbf{A}_p^{(\delta)}$ devient alors une matrice $\mathbf{A}_p^*$ tel que $\mathbf{A}_p^* = \begin{pmatrix} \mathbf{A}_p \\ \mathbf{C} \end{pmatrix}$. Cela revient au problème initial, sans contraintes sur le modèle. Le nombre de lignes linéairement indépendantes de la matrice $\mathbf{A}_p$ est égal à q, et $q < m$ donc $R(\mathbf{A}_p) < m$. De plus, la matrice contrainte $\mathbf{C}$ implique qu'une ligne de la matrice $\mathbf{A}_p$ devient une combinaison linéaire des autres, tel que $\forall h' = 1, \ldots, q$

$$\alpha \mathbf{A}_{p_{h'}} = - \sum_{h=1 (h \neq h')}^{q-1} \alpha \mathbf{A}_{p_h} + \sum_{k=1}^{p-1} \beta_k \mathbf{C}_k + \mathbf{C}_p \tag{A.5}$$

où α est un coefficient multiplicateur associé au vecteur ligne h', h de la matrice $\mathbf{A}_p$ et β est un coefficient multiplicateur associé au vecteur ligne k de la matrice $\mathbf{C}$. Il vient que $R(\mathbf{A}_p^*) = q + p - 1$, soit $R(\mathbf{A}_p^*) < m$. Le modèle initial, non contraint, n'est pas identifiable. Les bornes de la condition nécessaire A.4 indique le nombre minimum et maximum de probabilités de transition pouvant être fixées à zéro. Ces bornes sont nécessaires à l'obtention d'une matrice de plein rang.

Condition suffisante D'autre part, une condition suffisante à l'obtention d'un modèle identifiable est que le nombre de probabilités de transition fixés à zéro réponde à la contrainte suivante :

$$\sum_{h'=1}^{q} \sum_{h=1}^{p} (1 - \delta_{hh'}) \geq m - (q + \dim(p)) \tag{A.6}$$

Preuve La contrainte $\sum_{h'=1}^{q} \sum_{h=1}^{p} (1 - \delta_{hh'}) \geq m - (q + \dim(p))$ impose que le nombre minimal de lignes non nulles et non co-linéaires de la matrice $\mathbf{A}_p^{(\delta)}$, donc que le rang de la matrice $\mathbf{A}_p^{(\delta)}$, soit égal à la taille du vecteur $\mathbf{p}_.$. Selon le théorème de **Rouché-Fonténé**, cela signifie que la solution pour résoudre le système $\mathbf{A}_p^{(\delta)} \mathbf{p}_. = \mathbf{q}^{(\delta)}$ est unique. Les modèles respectant cette contrainte sont identifiables.

Égalité La quantité $m - (q + \dim(p))$ indique le nombre de vecteurs lignes non nuls de la matrice Δ. La matrice Δ étant une matrice diagonale, ces vecteurs sont linéairement indépendants. Lorsqu'il y a égalité, $\sum_{h'=1}^{q}, \sum_{h=1}^{p} (1 - \delta_{hh'}) = m - (q + \dim(p))$, les vecteurs non nuls de la matrice Δ sont également linéairement indépendants des autres vecteurs de la matrice $\mathbf{A}_p^{(\delta)}$. Le rang $R(\mathbf{A}_p^{(\delta)})$ correspond donc au rang de la matrice $R(\mathbf{A}_p^*)$ donné par l'équation A.5 et au nombre de vecteurs non nuls de la matrice Δ donné par la quantité $m - (q + \dim(p))$. Soit $R(\mathbf{A}_p^{(\delta)}) = R(\mathbf{A}_p^*) + (m - (q + \dim(p)))$. Le nombre de vecteurs non nuls de la matrice Δ donné par $(m - (q + \dim(p)))$ implique alors que la matrice $\mathbf{A}_p^{(\delta)}$ devienne une matrice de plein rang, dont le rang $R(\mathbf{A}_p^{(\delta)}) = m$, soit le nombre de paramètres inconnus. La solution est alors unique et le modèle correspondant est identifiable.

Inégalité Lorsqu'il y a inégalité stricte, $\sum_{h'=1}^{q} \sum_{h=1}^{p} (1 - \delta_{hh'}) > m - (q + \dim(p))$, la matrice $\mathbf{A}_p^{(\delta)}$ reste de plein rang. La quantité $m - (q + \dim(p))$ indique le nombre de vecteurs non nuls de la matrice Δ qui seront linéairement indépendant pour la matrice $\mathbf{A}_p^{(\delta)}$. Ce sont ces vecteurs qui permettent d'avoir une matrice $\mathbf{A}_p^{(\delta)}$ de plein rang alors les vecteurs non nuls supplémentaires de la matrice Δ deviennent des combinaisons linéaires des autres vecteurs de la matrice $\mathbf{A}_p^{(\delta)}$. Le rang de $\mathbf{A}_p^{(\delta)}$ reste alors équivalent à $R(\mathbf{A}_p^{(\delta)}) = R(\mathbf{A}_p^*) + (m - (q + \dim(p)))$, l'inégalité n'impliquant pas de nouveaux vecteurs indépendants nécessaires au changement du rang de la

matrice. La matrice $\mathbf{A}_p^{(\delta)}$ reste une matrice de plein rang, dont le rang $R(\mathbf{A}_p^{(\delta)}) = m$. La solution est unique et le modèle correspondant est identifiable.

Annexe B

Détection de modèles non-identifiables

Définition : Avec $\mathbf{p}$ connus, deux modèles $P(\boldsymbol{y};\mathbf{p}_{.},\mathbf{p},\delta)$ et $P(\boldsymbol{y};\mathbf{p}'_{.},\mathbf{p},\delta')$ sont dit **non identifiables** en $\mathbf{p}_{.}$ si il existe un couple de vecteurs de paramètres $(\mathbf{p}_{.},\mathbf{p}'_{.})$ tel que pour deux modèles (δ,δ'), les solutions soient identiques. Autrement dit : $\{(\delta,\delta')\in\Delta^2,\ \exists(\mathbf{p}_{.},\mathbf{p}'_{.})\in\mathcal{P}$ tel que $P(.;\mathbf{p}_{.},\mathbf{p},\delta)=P(.;\mathbf{p}'_{.},\mathbf{p},\delta')$ avec $\delta\neq\delta'$et $\mathbf{p}_{.}\neq\mathbf{p}'_{.}\}$

L'objectif est alors de résoudre le système en $p_{hh'}$ et $p'_{hh'}$:

$$\sum_{h=1}^{p}\delta_{hh'}p_{hh'}p_h=\sum_{h=1}^{p}\delta'_{hh'}p'_{hh'}p_h,\quad\forall h'=1,\dots,q \tag{B.1}$$

où les paramètres p_h sont supposés connus et identiques. Les deux modèles mis en comparaison étant connus, les paramètres $\delta_{hh'}$ sont également connus pour les deux modèles. Les paramètres inconnus de ce système sont alors les paramètres $\mathbf{P}_{.}=(\mathbf{p}_{.},\mathbf{p}_{.}{}')$. Les deux modèles comparés devant faire partie de l'ensemble de modèles $\boldsymbol{\Delta}$, ils répondent aux différentes contraintes définies dans la section 3.3. Soit :

$$s.c=\begin{cases}\sum_{h=1}^{p}p_h=1\\ \sum_{h'=1}^{q}p_{hh'}=1,\forall h,\\ \sum_{h'=1}^{q}p'_{hh'}=1,\forall h,\\ p_{hh'}=0,\forall h'\in I_h,\ \text{soit}\ \Delta_h\{\delta_{hh'}\}=\{0\},\\ p'_{hh'}=0,\forall h'\in I_h,\ \text{soit}\ \Delta_h\{\delta'_{hh'}\}=\{0\}.\end{cases} \tag{B.2}$$

où $\Delta_h\{\delta_{hh'}\}$ et $\Delta_h\{\delta'_{hh'}\}$ indiquent l'ensemble des paramètres fixés à zéro pour chacun des deux modèles. A ces contraintes, s'ajoutent des contraintes sur les paramètres $\mathbf{P}_{.}$ afin d'éviter de modifier les modèles comparés.

$$s.c \begin{cases} \delta_{hh'} = 1 \implies 0 < p_{hh'} <= 1 & \forall h = 1,\dots,p \ \forall h' = 1,\dots,q, \\ \delta'_{hh'} = 1 \implies 0 < p'_{hh'} <= 1 & \forall h = 1,\dots,p \ \forall h' = 1,\dots,q, \\ \delta_{hh'} = 0 \implies p_{hh'} = 0 & \forall h = 1,\dots,p \ \forall h' = 1,\dots,q, \\ \delta'_{hh'} = 0 \implies p'_{hh'} = 0 & \forall h = 1,\dots,p \ \forall h' = 1,\dots,q. \end{cases} \tag{B.3}$$

Les paramètres fixés à zéro sont indiqués par les modèles et les paramètres $\delta_{hh'} = 0$ et $\delta'_{hh'} = 0$. Il est alors nécessaire que les autres paramètres, (non fixés à zéro), où $\delta_{hh'} = 1$ et $\delta'_{hh'} = 1$ soient strictement positifs.

La problématique est alors d'identifier les modèles non identifiables. Pour cela 2 étapes sont nécessaires. Le système B.1 ainsi que les contraintes B.2 qui lui sont associées sont linéaires. De ce fait, la forme matricielle et ses propriétés peuvent être utilisées pour avoir une première information sur la solution du système. Lorsque le système possède une ou plusieurs solutions, il est nécessaire de vérifier qu'au moins une des solutions respectent les contraintes B.3. La vérification des contraintes B.3 est alors effectuée par l'étape 2. Lorsque le système ne possède aucune solution, l'étape 2 n'est pas nécessaire.

Etape 1 : Modélisation matricielle L'objectif de cette 1ère étape est d'obtenir une information sur la solution du système B.1 sous le respect des contraintes B.2. L'équation B.1 sous contraintes est un système linéaire d'inconnues $\mathbf{P}_{\cdot} = (\mathbf{p}_{\cdot}, \mathbf{p}'_{\cdot})$ tel que :

$$\sum_{h=1}^{p} \delta_{hh'} p_{hh'} p_h - \delta'_{hh'} p'_{hh'} p_h = 0, \quad \forall h' = 1,\dots,q. \tag{B.4}$$

De la même façon que dans l'annexe A, l'équation B.4 peut être mise sous forme matricielle tel que : $\mathbf{A_m P}_{\cdot} = \mathbf{q_m}$ où $\mathbf{P}_{\cdot} = (p_{11},\dots,p_{pq},p'_{11},\dots,p'_{pq})$ est le vecteur de paramètres inconnus de taille $m' = 2(pq)$ et $\mathbf{A_m}$ une matrice de taille qm', définie par :

$$\mathbf{A_m} = \begin{pmatrix} \delta_{11}p_1 & \dots & \delta_{1q}p_1 & \dots & \delta_{p1}p_p & \dots & \delta_{pq}p_p & \delta'_{11}(-p_1) & \dots & \delta'_{1q}(-p_1) & \dots & \delta'_{p1}(-p_p) & \dots & \delta'_{pq}(-p_p) \\ \vdots & \ddots & \delta_{1q}p_1 & \vdots & \vdots & \ddots & \delta_{pq}p_p & \vdots & \ddots & \vdots & \vdots & \delta'_{p1}(-p_p) & \ddots & \delta'_{pq}(-p_p) \\ \delta_{q1}p_1 & \dots & \delta_{1q}p_1 & \dots & \delta_{p1}p_p & \dots & \delta_{pq}p_p & \delta'_{11}(-p_1) & \dots & \delta'_{1q}(-p_1) & \dots & \delta'_{p1}(-p_p) & \dots & \delta'_{pq}(-p_p) \end{pmatrix}$$

Les contraintes $\sum_{h'=1}^{q} p_{hh'} = 1$ et $\sum_{h'=1}^{q} p'_{hh'} = 1$, $\forall h$, liées à l'équation 4.1 sont modélisées par la matrice $\mathbf{C_m}$ qui est de la forme :

$$\mathbf{C_m} = \begin{pmatrix} \mathbf{C} & 0 \\ 0 & \mathbf{C}' \end{pmatrix}$$

où les matrices $\mathbf{C}$ et $\mathbf{C}'$ correspondent aux matrices contraintes de chacun des deux modèles. La matrice contrainte $\mathbf{C}$ d'un modèle est présentée dans l'annexe A. Chacun des deux modèles est défini par sa matrice Δ, respectivement Δ', associée. La matrice Δ définissant les contraintes de type binaire imposées sur le modèle est également présentée en Annexe A. Les matrices Δ et Δ' sont regroupées dans une matrice Δ_m tel

que :

$$\Delta_m = \begin{pmatrix} \Delta & 0 \\ 0 & \Delta' \end{pmatrix}$$

Afin de comparer des modèles identifiables en paramètres, il est nécessaire que les matrices Δ et Δ' respectent les contraintes définies dans la section 3.3.2. L'équation B.4 sous contraintes est alors réécrite sous la forme : $\mathbf{A}_{\delta p}\mathbf{P}_{.} = \mathbf{q}_{\delta p}$,
avec $\mathbf{q}_{\delta p} = \left(q_1, \cdots, q_q, 1_1, \cdots, 1_{2p}, 0_1, \cdots, 0_{m'}\right)$

$$\underbrace{\begin{pmatrix} \mathbf{A_m} \\ \mathbf{C_m} \\ \Delta_m \end{pmatrix}}_{\mathbf{A}_{\delta p}} \underbrace{\begin{pmatrix} \mathbf{p}_{.} \\ \mathbf{p}'_{.} \end{pmatrix}}_{\mathbf{P}_{.}} = \underbrace{\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}}_{\mathbf{q}_{\delta p}} \tag{B.5}$$

En utilisant les propriétés des matrices, et notamment les propriétés liées au rang des matrices, il est possible de détecter si ce système possède une infinité de solutions, une solution unique ou aucune solution. De même que dans l'annexe A, les théorèmes de Rouché-Fonténé peuvent être utilisés.

Théorème 1 Selon le théorème de **Rouché-Fonténé** [102], un système d'équations linéaires à m' inconnues, de la forme $\mathbf{Ax} = \mathbf{b}$, possède une solution si et seulement si le rang de la matrice des coefficients $\mathbf{A}$ est égal à celui de la matrice augmentée $(\mathbf{A}|\mathbf{b})$.

Théorème 2 Selon le théorème de **Rouché-Fonténé** [102], lorsque le système d'équations linéaires admet une solution, celle ci est **unique** si et seulement si le rang de la matrice $\mathbf{A}$ est égal au nombre de paramètres inconnus (soit la taille du vecteur $\mathbf{P}_{.}$ dans notre cas).

Ces deux théorèmes impliquent que le système ne possède pas de solution lorsque le rang de la matrice $\mathbf{A}_{\delta p}$ est différent du rang de la matrice augmentée $(\mathbf{A}_{\delta p}|\mathbf{q}_{\delta p})$. Ces deux théorèmes, nous permettent donc d'avoir une première information pour détecter si les modèles sont identifiables ou non. En effet, par ces théorèmes, trois cas sont possibles :

Pas de solution : Le rang de la matrice $\mathbf{A}_{\delta p}$, noté $R(\mathbf{A}_{\delta_p})$, est inférieur au rang de la matrice augmentée $(\mathbf{A}_{\delta p}|\mathbf{q}_{\delta p})$: $R(\mathbf{A}_{\delta p}) < R(\mathbf{A}_{\delta p}|\mathbf{q}_{\delta p})$, la réponse est immédiate. Les deux modèles sont identifiables.

Une solution unique : $R(\mathbf{A}_{\delta p}) = R(\mathbf{A}_{\delta p}|\mathbf{q}_{\delta p})$ et $R(\mathbf{A}_{\delta p}) = m'$, il est nécessaire de passer à l'étape 2 de vérification des contraintes. Si la solution respecte les contraintes B.3, les deux modèles ne sont pas identifiables.

Infinité de solutions : $R(\mathbf{A}_{\delta p}) = R(\mathbf{A}_{\delta p}|\mathbf{q}_{\delta p})$ et $R(\mathbf{A}_{\delta p}) \neq m'$, a l'instar de la solution unique, il est nécessaire de passer à l'étape 2 de vérification des contraintes. Si l'une des solutions respecte les contraintes B.3, les deux modèles ne sont pas identifiables.

La forme matricielle ne prenant pas en compte les contraintes B.3, il est nécessaire de vérifier que celles-ci soient bien respectées lorsque le système possède une ou plusieurs solutions.

Etape 2 : Vérification des contraintes B.3 Afin de ne pas modifier les modèles comparés, seuls les paramètres $\delta_{hh'}$ indiquent les probabilités de transition fixées à zéro. Lorsqu'une ou plusieurs solutions existent, il est alors nécessaire que les probabilités de transition du vecteur $\mathbf{P}_{.}$ respectent les contraintes B.3. Cette vérification consiste à calculer la solution du système lorsque celle-ci est unique, ou à montrer qu'il existe une solution respectant les contraintes lorsqu'une infinité de solutions existent.

Exemple L'exemple suivant illustre la nécessité de vérifier que l'ensemble des probabilités de transition respectent les contraintes B.3.

Exemple : Dans cet exemple, les modèles de transition mis en compétition sont les modèles représentés par la figure B.1, δ_4 et δ_5 où $p = 3$ et $q = 5$.

·	·	·	p_{14}	p_{15}
p_{21}	p_{22}	·	·	p_{25}
·	·	p_{33}	·	·

(a) δ_4

·	·	·	p_{14}	p_{15}
p_{21}	·	·	·	p_{15}
·	p_{32}	p_{33}	·	·

(b) δ_5

FIGURE B.1 – Matrice des modèles de transition δ_4 et δ_5

Pour cet exemple, l'utilisation des rangs indique qu'il existe une infinité de solutions. $R(\mathbf{A}_{\delta p}) = R(\mathbf{A}_{\delta p}|\mathbf{q}_{\delta p}) = 29$ alors que $m' = 30$. Il est alors nécessaire de s'assurer que l'une des solutions respecte les contraintes B.3. Posant $\mathbf{p} = (0.7, 0.1, 0.2)$, la solution obtenue est :
$\mathbf{p}_1 = (0, 0, 0, r_1, 1 - r1)$, $\mathbf{p}_2 = (0, 0, 0, 0, 1)$, $\mathbf{p}_3 = (0, 0, 1, 0, 0)$
$\mathbf{p}'_1 = (0, 0, 0, r_1, 1 - r1)$, $\mathbf{p}'_2 = (0, 0, 0, 0, 1)$, $\mathbf{p}'_3 = (0, 0, 1, 0, 0)$
où r_1 est un paramètre pouvant prendre différentes valeurs impliquant l'infinité de solutions. Il est alors possible de constater que la solution retourne des vecteurs de paramètres $(\mathbf{p}_{.}, \mathbf{p}_{.}')$ identiques : $\mathbf{p}_{.} = \mathbf{p}'_{.}$. Ce qui implique une modification des modèles comparés et que les contraintes B.3 ne sont pas respectées. En effet, deux probabilités de transition ont été estimées à zéro alors qu'elles n'étaient pas imposées à zéros.

·	·	·	p_{14}	p_{15}
0	0	·	·	p_{25}
·	·	p_{33}	·	·

(a) δ_4

·	·	·	p_{14}	p_{15}
0	·	·	·	p_{25}
·	0	p_{33}	·	·

(b) δ_5

FIGURE B.2 – Matrice des modèles de transition δ_4 et δ_5

La figure B.2 présente les modèles modifiés. Les zéros indiqués en rouge, sont les zéros ajoutés par la solution. Les deux modèles sont alors identiques. D'autre part, l'ajout des zéros dans la solution implique que la contrainte $\sum_{h'=1}^{q}\sum_{h=1}^{p}(1-\delta_{hh'}) \geq m-(q+dim(p))$ ne soit plus respectée. Le modèle n'est alors plus identifiable en paramètres. Sans modification des modèles, le système n'a pas de solution, le paramètre $r1$ impliquant l'infinité de solution ne participant pas à la modification des modèles. Les deux modèles mis en compétition sont donc **identifiables** sous le respect des contraintes B.3.

Résumé

Dans le domaine de la comparaison d'assurances en ligne, les données évoluent constamment, impliquant certaines difficultés pour les exploiter. En effet, la plupart des méthodes d'apprentissage standards, comme la classification supervisée, nécessitent des descripteurs de données identiques pour les échantillons d'apprentissage et de test. Or, les formulaires en lignes d'où proviennent les données sont régulièrement modifiés, impliquant de travailler avec une faible quantité de données. L'objectif est alors d'utiliser les données obtenues avant la modification des descripteurs pour générer de nouveaux échantillons et augmenter la taille des échantillons observés après la modification. Nous proposons donc d'effectuer un transfert de connaissances entre les données observées avant et après la modification des variables. Les données étant observées soit avant, soit après la modification de la variable, entraînent un problème de données manquantes où les liens entre les descripteurs avant et après la modification sont totalement inconnus. Une modélisation probabiliste du problème, modélisant la loi jointe de la variable avant et après la modification de ses descripteurs est proposée. Le problème revient alors à un problème d'estimation dans un graphe où le modèle n'est pas identifiable. L'identifiabilité du modèle est assurée par des contraintes métiers et techniques, amenant à travailler avec un ensemble réduit de modèles très parcimonieux. Deux méthodes d'estimation des paramètres reposant sur des algorithmes EM sont proposées : une estimation par vraisemblance profilée et une estimation jointe des paramètres. L'ensemble de modèles amène à une étape de sélection de modèle, effectuée selon deux critères : un critère asymptotique et un critère non asymptotique reposant sur l'analyse bayésienne, comprenant une stratégie d'échantillonnage préférentiel combinée à un algorithme de Gibbs. Pour obtenir la méthode ayant le meilleur compromis "résultats-temps de calcul", une recherche exhaustive (EXsearch) et une recherche non-exhaustive (AGsearch) sont comparées. La recherche AGsearch, basée sur un algorithme génétique, combine à la fois l'estimation (problème continu) et la sélection de modèle (problème combinatoire). La thèse se termine par une comparaison des méthodes et critères proposés afin d'obtenir une stratégie optimale, puis par une application sur des données réelles.

Mots clés : transfert de connaissance, sélection de modèle, algorithme génétique

Abstract

In the online insurance comparison field, data constantly evolve, implying some difficulties to exploit them. Indeed, most of the classical learning methods, as supervised classification, require data descriptors equal to both learning and test samples. Online forms where data come from are often changed. These constant modifications of data descriptors lead us to work with the small amount of data and make analysis more complex. So, the goal is to use data generated before the feature descriptors modification. By doing so, we generate new samples and increase the size of the observed sample after the descriptors modification. We intend to perform a learning transfer between observed data before and after features modification. Data are observed either before, or after feature modification which bring a problem of missing data. Also, the links between data descriptors of the feature before and after the modification are totally unknown. A probabilistic modelling of the problem has been suggested to modelize the joint distribution of the feature before and after the modification of the data descriptors. The problem becomes an estimation problem in a graph where the model is unidentifiable. Some business and technical constraints ensure the identifiability of the model and we have to work with a reduced set of very parsimonious models. Two methods of estimation rely on EM algorithms have been intented. The first one is an estimation by profile likelihood and the second one is a join estimation of parameters. The constraints set lead us to work with a set of models. A model selection step is required. For this step, two criterium are proposed: an asymptotic criterium and a non-asymptotic criterium rely on Bayesian analysis which includes an importance sampling combined with Gibbs algorithm. To have an optimal method for both results and execution time, two research strategies are suggested. The first strategy (EXsearch) is an exhaustive search and the second strategy (AGsearch) is a non-exhaustive search based on genetic algorithm, combining both estimation (continuous problem) and selection (combinatorial problem). This thesis finishes with a comparison of methods and criteria proposed to detect the optimal strategy in a business framework and with an application on real data.

Keywords: transfer learning, models selection, genetic algortithms